BRL CR 334

# BRL



CONTRACT REPORT NO. 334

# NON LINEAR DYNAMIC ANALYSIS OF FLAT LAMINATED PLATES BY THE FINITE-ELEMENT METHOD



Prepared by

University of Illinois
Aeronautical and Astronautical
Engineering Department
Urbana, Illinois

March 1977

Approved for public release; distribution unlimited.

USA ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
USA BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | |
|---|---|
| 1. REPORT NUMBER<br>Contract Report Number 334 | 2. GOVT ACCESSION NO. |
| 3. RECIPIENT'S CATALOG NUMBER | |
| 4. TITLE (and Subtitle)<br>Non Linear Dynamic Analysis of Flat Laminated Plates by the Finite-Element Method | 5. TYPE OF REPORT & PERIOD COVERED<br>BRL Contractor Report<br>1 Jan 75 to 30 Jun 76 |
| | 6. PERFORMING ORG. REPORT NUMBER<br>None |
| 7. AUTHOR(s)<br>A. R. Zak | 8. CONTRACT OR GRANT NUMBER(s)<br>DAAD05-73-C-0197 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Aeronautical and Astronautical Engr. Dept<br>University of Illinois<br>Urbana, Illinois | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>1T161702A33H, 002AJ<br>61702A |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>USA Ballistic Research Laboratory<br>Aberdeen Proving Ground, Maryland 21005 | 12. REPORT DATE<br>MARCH 1977 |
| | 13. NUMBER OF PAGES<br>83 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)<br>US Army Materiel & Readiness Command<br>5001 Eisenhower Avenue<br>Alexandria, VA 22333 | 15. SECURITY CLASS. (of this report)<br>Unclassified |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES



19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Anisotropic, Analysis, Finite-Element, Three Dimensional Deformations, Dynamic Response

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A finite-element structural model has been developed for the dynamic analysis of laminated, thick plates. The model uses constant thickness quadrilateral elements to represent the shape of the plate and these elements are stacked in the thickness direction to represent the desired material layers. The analysis allows for orthotropic material properties of each layer as well as for elastic-plastic material response. Nonlinear strain displacement relations are used in the formulation to represent large, transverse plate deflections.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

The finite-element model is used to prepare a computer program for the numerical calculations. Two versions of the program have been prepared, which correspond to two different time integration numerical methods. These methods include finite-difference and predictor-corrector techniques. The computer programs are designed for time and space dependent pressure loads to be applied to one surface of the plate. However, the programs could be used for other loading conditions by changing one subroutine.

TABLE OF CONTENTS

# I INTRODUCTION

The purpose of this investigation has been to develop a finite-element model and a computer program for the dynamic analysis of flat, laminated plate structures. The finite-element model uses the quadrilateral to define the shape of the element in the plane of the plate and the thickness direction is represented by arranging a number of these elements to describe the necessary number of material layers. Each material layer can be assigned different material properties and the computer program is designed to allow for maximum of twelve layers, however, this can easily be increased or decreased. In the detail development of the model each quadrilateral element is further subdivided into four triangular elements[1,2].

The model in this analysis is nonlinear since it allows for material yield effects and for large plate deflections. The large deflection effects are introduced by assuming that the transverse displacement is large compared to the two displacements in the plane of the plate. This leads to second order terms in the strain-displacement relations. The yield effects are introduced because it is intended that the application of this model will be in the situation where the loads are large enough to produce stresses beyond the elastic limit. Consequently, the present analysis allows for elastic-plastic material properties which are introduced into the model by checking the yield for each element and when the yield is exceeded, the state of stress in the element is adjusted by using the plasticity flow rule. This is done by checking for yield at each time interval used in the numerical integration of the dynamic equations.

The dynamic equations for the plate are obtained by lumping the mass of the plate into the nodal points of the finite-element model. This leads to a set of concentrated masses distributed in the plane of the plate and in the thickness direction. The solution to these equations is obtained numerically in the computer program. In the preliminary version of the program three different integration techniques are investigated. These include an iterative approach, a finite-difference method, and a predictor-corrector method. Based on some results obtained for simple dynamic problems, it was found that the fastest methods were the finite-difference and the predictor-corrector with the iterative approach being

---

[1]Zienkiewicz, O. C., "The Finite Element Method in Engineering Science," McGraw-Hill Publishing Company, London, 1971.

[2]Przemieniecki, J. S., "Theory of Matrix Structural Analysis," McGraw-Hill Book Company, New York, 1968.

the slowest. This of course does not imply that the same results would be obtained for all dynamic problems, however, in the present investigation it was necessary to limit the number of versions of the computer program and, therefore, it was decided to prepare two final versions using the finite-difference and the predictor-corrector methods of integration.

The analysis which has been developed is quite general and corresponding computer program could be used for a wide class of problems and loading conditions. Various loading conditions can be generated by supplying user subroutine to define the load in space and time. However, the present versions of the program is setup for particular load which involves a distributed pressure on one surface of the plate. This pressure can be specified as a function of time and of the inplane coordinates.

## II THEORETICAL DEVELOPMENT

### Finite-Element Model

The present analysis is intended to handle thick, plate-like structures, which are composed of different material layers in the thickness directions. The plane of the plate is parallel to the x-y coordinates and the thickness is represented by the z direction as shown in Figure 1. The analysis is developed in terms of these Cartesian coordinates. The shape of the finite elements is defined by a general quadrilateral in the x-y plane and each element has a constant thickness in the z direction with the limitation that no element will contain more than one material. Each layer of the material can therefore be represented by one or more elements in the thickness direction.

A typical element is shown in Figure 2. Each quadrilateral element is subdivided into four triangular elements with the node in the center being a temporary that will later be eliminated by static condensation. A general triangular element is shown in Figure 3. The nodal numbering system for each quadrilateral element is also given. There are three degrees of freedom at each node.

The first step in the finite element analysis is to assume a suitable displacement function over each triangular element. The displacement functions chosen for this analysis are

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y + z\ (\beta_1 + \beta_2 x + \beta_3 y)$$

$$v = \alpha_4 + \alpha_5 x + \alpha_6 y + z\ (\beta_4 + \beta_5 x + \beta_6 y)$$

$$w = \alpha_7 + \alpha_8 x + \alpha_9 y + z\ (\beta_7 + \beta_8 x + \beta_9 y) \qquad (1)$$

where $\alpha_i$, i = 1, 9, and $\beta_i$, i = 1, 9, are unknown coefficients. This displacement function is linear in planes parallel to the x-y plane and varies linearly with z through the thickness of each element. Writing Equations (1) at each of the six nodes of a triangular element results in eighteen equations which can be written in matrix form as

$$\{\delta\} = [C]\ \{\alpha\} \tag{2}$$

where $\{\delta\}$ denotes the nodal displacements, $[C]$ is a known constant matrix depending on the local nodal coordinates, and $\{\alpha\}$ is the vector of unknown coefficient $\alpha_i$ and $\beta_i$ defined in Equations (1). Solving for $\{\alpha\}$ results in

$$\{\alpha\} = [C]^{-1}\ \{\delta\} \tag{3}$$

It may be noted at this time that as a consequence of choosing the displacement variations as given by Equations (1), the inverse of the [C] matrix in equation (3) can be performed analytically thereby leading to an appreciable saving in numerical work. The present analysis is designed to account for nonlinear effects arising from large deflections of the plate. In order to account for these deflections analytically, it is assumed that the deflections and rotations out of the plane of the plate are larger compared to those in the plane of the plate. The resulting non-linear strain-displacement relations are therefore given by

$$\begin{aligned}
\varepsilon_{11} &= \frac{\partial u}{\partial x} + \frac{1}{2}\left(\frac{\partial w}{\partial x}\right)^2 \\
\varepsilon_{22} &= \frac{\partial v}{\partial y} + \frac{1}{2}\left(\frac{\partial w}{\partial y}\right)^2 \\
\varepsilon_{33} &= \frac{\partial w}{\partial z} + \frac{1}{2}\left[\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2\right] \\
\varepsilon_{12} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \left(\frac{\partial w}{\partial x}\right)\left(\frac{\partial w}{\partial y}\right) \\
\varepsilon_{23} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\
\varepsilon_{31} &= \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}
\end{aligned} \tag{4}$$

Equations (4) may be written in term of linear and nonlinear terms. Using matrix notation

$$\{\varepsilon\} = \{\varepsilon_o\} + \{\varepsilon_L\} \tag{5}$$

where

$$\{\varepsilon_o\} = \left[\frac{\partial u}{\partial x},\ \frac{\partial v}{\partial y},\ \frac{\partial w}{\partial z},\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x},\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y},\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right]^T \tag{6}$$

$$\begin{aligned}
\{\varepsilon_L\} = \Big[ &\frac{1}{2}\left(\frac{\partial w}{\partial x}\right)^2,\ \frac{1}{2}\left(\frac{\partial w}{\partial y}\right)^2,\ \frac{1}{2}\left(\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2\right), \\
&\frac{\partial w}{\partial x}\cdot\frac{\partial w}{\partial y},\ 0,\ 0\Big]^T
\end{aligned} \tag{7}$$

<u>Dynamic Equations</u>

The next step in the finite element model derivation is the calculation of the virtual change in the internal work of the structure due to virtual changes in the nodal displacements. The virtual change in the internal work due to the stresses is given by an integral over the volume of

a particular element. This can be represented by

$$dW_I = \int_V d\{\varepsilon\}^T \{\sigma\} \, dV \tag{8}$$

where $dW_I$ represents the virtual change in the internal work, $d\{\varepsilon\}^T$ is the transpose of the matrix representing the virgual changes in the strains, and $\{\sigma\}$ is the matrix of the stresses. The quantity V represents the volume of the element. The first step in obtaining the integral in Equation (8) is to derive the expression for the virtual changes in the strains. Using Equation (5) written in two parts

$$d\{\varepsilon\} = d\{\varepsilon_o\} + d\{\varepsilon_L\} \tag{9}$$

Consider first the linear strain component of virtual strain $d\{\varepsilon_o\}$. By using the definition of $\{\varepsilon_o\}$ given by Equation (6) and the displacement function in Equations (1) it is possible to write

$$\{\varepsilon_o\} = [Q] \{\alpha\} \tag{10}$$

where [Q] is a matrix whose elements are functions of x, y and z. By combining Equations (3) and (10) it follows

$$\{\varepsilon_o\} = [B_o] \{\delta\} \tag{11}$$

where the $[B_o]$ matrix is defined as

$$[B_o] = [Q] [C]^{-1} \tag{12}$$

By introducing the virtual change of nodal displacements it follows that

$$d\{\varepsilon_o\} = [B_o] \, d\{\delta\} \tag{13}$$

where $d\{\delta\}$ are changes in the nodal displacements.

Consider now the virtual change in the nonlinear component of strain $d\{\varepsilon_L\}$. Consider Equation (7) which can be rewritten in a different form

$$\{\varepsilon_L\} = \frac{1}{2} [A] \{\theta\} \tag{14}$$

where the new matrices in Equation (14) are defined as follows

$$[A] = \begin{bmatrix} \frac{\partial w}{\partial x} & 0 & 0 & 0 \\ 0 & \frac{\partial w}{\partial y} & 0 & 0 \\ 0 & 0 & \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial y} & \frac{\partial w}{\partial x} & 0 & 0 \end{bmatrix}$$

$$\{\theta\} = [\ \frac{\partial w}{\partial x}\ ,\ \frac{\partial w}{\partial y}\ ,\ \frac{\partial u}{\partial z}\ ,\ \frac{\partial v}{\partial z}\ ]^T \tag{15}$$

It may be noted that using Equations (15) the strain $\{\varepsilon_L\}$ has only four elements while the original definition of $\{\varepsilon_L\}$, in Equation (5), has six elements. However, it may be noted from the original definition that the last two elements $\{\varepsilon_L\}$ were identically zero and consequently $\{\varepsilon_L\}$ in Equation (13) represents only the non zero part of the original strains. The virtual change in the strain $\{\varepsilon_L\}$ can be written as

$$d\{\varepsilon_L\} = \frac{1}{2}\, d[A]\ \{\theta\} + \frac{1}{2}\, [A]\ d\{\theta\} \tag{16}$$

However, by using Equations (15) it can be shown that

$$d[A]\ \{\theta\} = [A]\ d\{\theta\} \tag{17}$$

therefore

$$d\{\varepsilon_L\} = [A]\ d\{\theta\} \tag{18}$$

By using the definition of $d\{\theta\}$ it is possible to write

$$d\{\theta\} = [X]\ d\{\delta\} \tag{19}$$

It may be noted at this stage that the matrix [A] contains the coordinates x, y, z and some of the unknown coefficient $\alpha_i$ and $\beta_i$ defined in Equations (1). During the programming of this analysis for numerical calculations, it was found convenient to write Equation (19) in slightly different form

$$d\{\varepsilon_L\} = [Z]\ [\bar{C}]\ d\{\delta\} \tag{20}$$

where obviously

$$[Z]\ [\bar{C}] = [A]\ [X] \tag{21}$$

The matrix $[\bar{C}]$ in Equation (21) is actually related to the matrix $[C]^{-1}$ defined in Equation (3) and [Z] contains the variables x, y, z and the $\alpha_i$ and $\beta_i$ coefficients.

Returning to Equation (8), the internal work term can be written as

$$dW_I = d\{\delta\}^T \int_V [B_o]^T\ \{\sigma\}\ dV$$
$$+\ d\{\delta\}^T\ [\bar{C}]^T \int_V [Z]^T\ \{\sigma\}\ dV \tag{22}$$

The stress matrix $\{\sigma\}$ is defined as follows

$$\{\sigma\} = [\ \sigma_{11},\ \sigma_{22},\ \sigma_{33},\ \sigma_{12},\ \sigma_{23},\ \sigma_{31}]^T \tag{23}$$

However, it may be noted that in the second integral in Equation (22) only the first four stresses from Equation (23) are needed. By combining the two integrals in Equation (22) it is possible to write

$$\delta W_I = d\{\delta\}^T \int_V [B]^T \{\sigma\} \, dV \quad (24)$$

where $\{\sigma\}$ is defined according to Equation (23).

In addition to the work done by internal forces in each finite element, Equation (24), there is work done by the forces at the eight corners. By denoting these forces by the matrix $\{f\}$, the external virtual work done is obtained in the form

$$dW_E = d\{\delta\}^T \{f\} \quad (25)$$

Since the external forces on each element are in equilibrium with the stresses within the element it follows from the principle of virtual work that

$$\delta W_I + \delta W_E = 0 \quad (26)$$

and therefore

$$\{f\} = - \int_V [B]^T \{\sigma\} \, dV \quad (27)$$

The next step in the analysis is to obtain the matrix equilibrium equation for the total structure. Before this can be done, it is necessary to introduce the inertia effects. In the present analysis the mass of the structure is lumped at the nodes. Each side of the triangular element shown in Figure 3 is bisected and these center points are joined to the center node which defines the apex of the four triangular elements previously defined in Figure 2. By performing this step the original quadrilateral has been divided into four smaller quadrilaterals and each of these contains two of the eight nodes of the element. The mass of each smaller quadrilateral portion of the element is obtained by multiplying the area of the quadrilateral by the thickness of the element in the z direction and by the material density and this mass is distributed equally to the two nodes. Following this procedure, the mass matrix for the total structure is obtained by summing masses at each structure node from the adjacent elements. The resulting matrix is diagonal.

The next step of the analysis is the summation of forces at each node of the structure. The forces acting on any node from the adjacent elements are obtained from Equation (27). The sum of these forces can be represented for the total structure by a matrix which will be denoted by $\{F_I\}$. The additional forces which affect the system are the external forces. The present model will allow for these to be surface pressure forces, which vary with space and time. Using the principle of virtual work these pressure loads are transformed to an equivalent set of concentrated nodal forces. These forces will be denoted by a matrix $\{F_E\}$. Using the two forces the matrix equilibrium equation can be written in the form

$$[M] \{\ddot{\Delta}\} = \{F_I\} + \{F_E\} \quad (28)$$

where [M] is the mass matrix, and $\{\Delta\}$ represents the global displacement matrix with the double dot denoting the second time derivative. In the present investigation the solution to Equation (28) is obtained using numerical integration methods which will be described in the next section

of this report.

## III NUMERICAL ANALYSIS

The solution of Equation (28) is obtained by using two different numerical integration techniques. These two different methods include a finite-difference approach and a predictor-corrector method. Both of these methods were used to prepare two different versions of the computer program. A brief description of both of these methods is given below.

### Finite Difference Method

The numerical solution of Equation (28) by the finite difference method involves the replacement of the time derivatives by finite difference equivalents [3,4,5]. In this approach the time history is divided into discreet intervals whose length will be denoted by h. For convenience any general element of the displacement vector $\{\Delta\}$ at a given time will be represented by the quantity x with a subscript defining the time interval. From the general theory of kinematics, the velocity and displacement relations are written for nth and (n+1) th time intervals as follows

$$\dot{x}_n = \dot{x}_{n-1} + \frac{h}{2}(\ddot{x}_{n-1} + \ddot{x}_n) \tag{29}$$

$$x_n = x_{n-1} + h\dot{x}_{n-1} + (\frac{1}{2} - \beta)h^2\ddot{x}_{n-1} + \beta h^2\ddot{x}_n \tag{30}$$

$$\dot{x}_{n+1} = \dot{x}_n + \frac{h}{2}\ddot{x}_n + \frac{h}{2}\ddot{x}_{n+1} \tag{31}$$

$$x_{n+1} = x_n + h\dot{x}_n + (\frac{1}{2} - \beta)h^2\ddot{x}_n + \beta h^2\ddot{x}_{n+1} \tag{32}$$

where $\ddot{x}$, $\dot{x}$, and x represent acceleration, velocity and displacements at the time intervals denoted by the subscript.

Equations (29) and (31) mean that the velocity at the end of the interval is equal to the sum of the velocity at the beginning of the interval and the product of the length of the interval, h, and the average of the accelerations at the beginning and end of the interval. Equations (30) and (32) are obtained by integrating Equations (29) and (31) and introducing

---

[3] Newmark, Nathan, M., "A Method of Computation for Structural Dynamics," Journal of the Engineering Mechanics Division, ASCE, July, 1959.

[4] Chan, S. P., Cox, H. L., and Benfield, W. A., "Transient Analysis of Forced Vibrations of Complex Structural-Mechanical Systems," Journal of the Royal Aeronautical Society, July, 1962.

[5] Wu, R., and Witmer, E. A., "Nonlinear Transient Responses of Structures by the Spatial Finite Element Method," AIAA J., August, 1973.

the acceleration parameter, $\beta$, to express the acceleration at the beginning and end of the interval. For example, $\beta = \frac{1}{4}$ means that the acceleration during the interval is constant and is equal to the mean of the accelerations at the beginning and end of the time interval. The $\beta$ parameter is known as the generalized acceleration parameter and its value is chosen in the numerical calculations to insure convergence and stability of the numerical results. According to previously published results[3] it has been found that in order to insure stability and convergence the value of this parameter should be kept in the range of $0 < \beta < 1/4$. The different values of this parameter are suitable for different types of dynamic problems, however, when $\beta = 1/4$ the stability limit on the integration time interval is infinite. Consequently, this value of $\beta$ should be suitable in most dynamic problems.

At $t = (n+1)h$, $nh$, and $(n-1)h$, respectively, the equation of motion, Equation (28), becomes

$$
\begin{aligned}
[M]\{\ddot{\Delta}\}_{n+1} - \{F_I\}_{n+1} &= \{F_E\}_{n+1} \\
[M]\{\ddot{\Delta}\}_{n} - \{F_I\}_{n} &= \{F_E\}_{n} \\
[M]\{\ddot{\Delta}\}_{n-1} - \{F_I\}_{n-1} &= \{F_E\}_{n-1}
\end{aligned}
\tag{33}
$$

By combining Equations (29) to (33) it can be shown, see Appendix A, that this leads to the following difference equation for the displacement at the n+1 time interval

$$
\begin{aligned}
[M]\{\Delta\}_{n+1} = {} & 2[M]\{\Delta\}_n - [M]\{\Delta\}_{n-1} \\
& + \beta h^2(\{F_I\}_{n+1} + (1/\beta - 2)\{F_I\}_n + \{F_I\}_{n-1}) \\
& + \beta h^2(\{F_E\}_{n+1} + (1/\beta - 2)\{F_E\}_n + \{F_E\}_{n-1})
\end{aligned}
\tag{34}
$$

Equation (34) is used in the computer program to calculate the displacement at the time interval n+1 from the response at the two previous time intervals n and n-1. In applying Equation (34) it is assumed that the internal force $\{F_I\}_{n+1}$ is dependant on the stresses calculated at the time interval n. The physical interpretation is that the analysis treats the internal force as a step function rather than as a continuous force. This can be illustrated by a curve shown in Figure 4, which represents a force constant over each time interval. Obviously this is an approximation, however, as the time interval decreases this curve will approach a continously varying function.

It may be noted that Equation (34) can not be applied to the first time interval since $\{\Delta\}_{n-1}$ does not exist. Consequently, in the beginning of this solution procedure the displacement at the end of the first interval must be formulated in terms of the initial velocity and initial

displacement since these are the only known quantities. It is possible to use the following starting procedure

$$[M]\ \{\Delta\}_1 + \beta h^2\ (-\{F_I\}_{,1}) = [M]\ \{\Delta\}_o$$

$$- (\tfrac{1}{2} - \beta)h^2\ (-\{F_I\}_{,o}) + [M]h\ \{\dot{\Delta}\}_o$$

$$+ \beta h^2\ \{F_E\}_{,1} + (\tfrac{1}{2} - \beta)[I]h^2\ \{F_E\}_{,o} \tag{35}$$

where [I] is the unit matrix. Equation (35) is derived in a manner similar to Equation (34).

Thus, the general procedure is to start with Equation (35) in order to obtain the displacements at the end of the first time interval and subsequently use Equation (34) to obtain the displacements at later times. (Note that when $\beta = 0$, this method reduces to the central finite difference method.)

Predictor-Corrector Method

In general, predictor-corrector methods involve using a truncated formula to 'predict' the value of the unknown and then applying a more accurate 'corrector' formula to provide successive improvements.

The predictor-corrector subroutine used in this analysis is named DHPCG (Hamming's Predictor-Corrector Method) and is from the IBM- Scientific Subroutine Package which also gives a detailed explanation of the procedure[6]. In brief, Hamming's Predictor-Corrector method gives an approximate numerical solution to a first order linear ordinary differential equation with given initial conditions. It is a stable fourth order procedure in which the user may vary the step-size. DHPCG also estimates the local truncation error.

Elastic-Plastic Analysis

The present analysis allows for elastic-plastic response of the structure. Because of practical considerations each finite element is assumed to be either elastic or plastic. However, the stress strain relation which can be written in a matrix form as

$$\{\sigma\} = [D]\ \{\varepsilon\} \tag{36}$$

---

[6] Ralston, A., and Wilf, H. S., "Mathematical Methods for Digital Computers," Wiley, New York, 1960, p. 95-109.

predicts that the stresses will vary within each element since $\{\varepsilon\}$, as given by Equation (5) is a function of the coordinates. In order to have one representation yield criterion for each element, a numerical procedure was adopted, which is used in Equation (36), which averages the stresses over each element. These average stresses were used in evaluating the internal nodal forces in Equation (27).

The Mises-Hencky yield criterion[5] is used to determine if plastic flow has occured in any given element. The stresses and strains, which here will be denoted by indicial notation at time $t_n$ and the displacements at time $t_{n+1}$ are known. From this information the stress increment at time $t_{n+1}$ is calculated from the elastic constitutive relations,

that is,

$$(\Delta\sigma_{ij})^T_{n+1} = \frac{E}{1+\nu} \left| (\Delta\varepsilon_{ij})_{n+1} + \frac{\nu}{1-2\nu} (\Delta\varepsilon_{kk})_{n+1} \delta_{ij} \right| \tag{37}$$

The total stress at $t_{n+1}$ is

$$(\sigma_{ij})^T_{n+1} = (\sigma_{ij})_n + (\Delta\sigma_{ij})^T_{n+1} \tag{38}$$

where the superscript 'T' denotes a trial stress state and $(\sigma_{ij})^T_{n+1}$ is the total trial stress at time $t_{n+1}$, $(\sigma_{ij})_n$ is the total stress at time $t_n$, and $(\Delta\sigma_{ij})^T_{n+1}$ is the trial stress increment at time $t_{n+1}$.

The Mises-Hencky yield function is given by

$$\Phi = S_{ij}\, S_{ij} - \frac{2}{3}\, \sigma_y{}^2 \tag{39}$$

where $S_{ij}$ are the deviatoric components of stress, that is,

$$S_{ij} = \sigma_{ij} - \frac{1}{3}\, \sigma_{kk}\, \delta_{ij} \tag{40}$$

and $\sigma_y$ is the known uniaxial yield stress of the material.

Substituting Equation (37) into Equation (38) gives

$$\Phi^T_{n+1} = (S_{ij})^T_{n+1} (S_{ij})^T_{n+1} - \frac{2}{3}\, \sigma_y{}^2 \tag{41}$$

If $\Phi^T_{n+1} < 0$ then the trial stress state is in the elastic region and no plastic flow has occured. In this case the total stress is given by Equation (38) or

$$(\sigma_{ij})_{n+1} = (\sigma_{ij})_n + (\Delta\sigma_{ij})_{n+1} \tag{42}$$

If $\Phi^T_{n+1} \geq 0$ then plastic flow has occured and the stress increments is not totally elastic as was assumed. The stress state must lie on the yield surface as specified by the theory of perfect plasticity. To calculate the new stress state, the strain increment is broken down into elastic and plastic components, that is,

$$(\Delta\varepsilon_{ij})_{n+1} = (\Delta\varepsilon_{ij})^{e}_{n+1} + (\Delta\varepsilon_{ij})^{p}_{n+1} \tag{43}$$

where the superscripts 'e' and 'p' denote the elastic and plastic components, respectively. From the incompressibility condition of plasticity and by the flow rule, the plastic strain increment is given by

$$(\Delta\varepsilon_{ij})^{p}_{n+1} = (S_{ij})^{T}_{n+1}\ \tilde{\lambda} \tag{44}$$

where $\tilde{\lambda}$ is a real nonnegative scalar quantity.

Combining Equation (44) with Equation (37) the stress increment is

$$(\Delta\sigma_{ij})_{n+1} = \frac{E}{1+\nu}\left[\Delta\varepsilon_{ij} + \frac{\nu}{1-2\nu}\ \Delta\varepsilon_{kk}\ \delta_{ij} - (S_{ij})^{T}_{n+1}\ \tilde{\lambda}\right] \tag{45}$$

and the actual stress at time $t_{n+1}$

$$(\sigma_{ij})_{n+1} = (\sigma_{ij})^{T}_{n+1} - (S_{ij})^{T}_{n+1}\ \lambda^{*} \tag{46}$$

where

$$\lambda^{*} = \tilde{\lambda}E/1+\nu$$

The quantity $\lambda^{*}$ may be determined from the fact that $(\sigma_{ij})_{n+1}$ in Equation (46) must satisfy the yield criterion that is $\Phi = 0$. Substituting Equation (46) into Equation (39) and solving for $\lambda^{*}$ gives

$$\lambda^{*} = \frac{C}{B+\sqrt{B^{2} - AC}} \tag{47}$$

where for convenience the following parameters were introduced

$$A = (S_{ij})^{T}_{n+1}\ (S_{ij})^{T}_{n+1}$$

$$B = (S_{ij})^{T}_{n+1}\ (\sigma_{ij})^{T}_{n+1}$$

$$C = (\sigma_{ij})^{T}_{n+1}\ (\sigma_{ij})^{T}_{n+1} - \frac{1}{3}\ (\sigma_{kk})^{T\,2} - \frac{2}{3}\ \sigma_{y}{}^{2} = \Phi^{T}_{n+1} \tag{48}$$

## Computer Program Inputs

The computer programs for the two different versions of integration have been designed to have the same type of input cards. The description of the input cards is given in Appendix B. The two versions of the computer program are quite similar as seen from the flow charts shown in Figures 5 and 6, which are for the integration finite-difference and predictor-corrector methods respectively. These charts show the main subroutines

whereas some third level subroutines have been left out for clarity. It can be seen that all of these programs utilize an automatic mesh generation which is done in MESH and POINTS. The differences are in the second part of the program, which starts with the subroutine DIFF. In the finite-difference approach, this subroutine is in the Loop 1, which is on the time interval. Inside of DIFF there is a second loop called Loop 2. In predictor-corrector method Loop 2 checks for yield only. In the predictor-corrector method the subroutine DIFF is not in a time loop since the subroutine DHPCG automatically cycles through all the time intervals.

In order to illustrate the actual program arrangement, a listing is given in Appendix C of the program corresponding to the finite-difference method. The plate load is specified by a Subroutine DISFOR which has to be supplied by the user of the program. This subroutine converts the distributed load to equivalent concentrated nodal forces. This subroutine is called for each quadrilateral element in the loaded plane of the plate and it calculates the value of the distributed load FF(I), I = 1, 4) for each corner of the quadrilateral and for a given time parameter TIME. The conversion to concentrated forces is then automatic.

## Solution of Laminated Plate

In order to check the accuracy of the dynamic plate analysis it was decided to apply the finite-difference approach to the analysis of the dynamic response of a laminated plate which was tested experimentally at the Ballistic Research Laboratories. The results of this test were reported recently[7] where comparison was also made with numerical calculations performed with a hydrodynamic code called HEMP.

The laminated plate consists of three layers, two of steel and one of aluminum. The plate has a rectangular shape 250 x 500 mm, and is supported at the four corners by resting on four posts. The plate was loaded by exploding a circular explosive at the center of the plate and the measurements consisted of optical measurement of the plate deflection on the side away from the explosive.

In order to model the experimental configuration by the finite-element plate program, it was necessary only to consider one quarter of the plate because of symmetry. Each layer of the plate was represented by eight rectangular elements as shown in Figure 7. Although this is a relatively coarse grid, it was considered to be sufficient in order to determine whether or not if the present analysis would produce answers of the correct order of magnitude. The pressure load was applied at the four nodes closest to the center of the plate by converting the pressure to time-dependant concentrated forces. These pressures were supplied by the BRL and were numerically obtained from the HEMP hydrodynamic program[7]. The four points at which the loads were applied are indicated in Figure 7.

[7] Majerus, J. N., and Karpp, R. R., "Dynamic Behavior of Multi-Layered Plate Due to an Intense Impulsive Load," Proceedings of the Second International Conference on Mechanical Behavior of Materials," Boston, Mass., August, 1976.

Since a relatively coarse finite-element grid was used this force distribution can only be considered as a rough approximation.

The numerical calculations were performed using a time interval of 0.25μs, and the calculations were performed up to 60μs. At this time the load was almost equal to zero since its decay is quite rapid. The deflection of the plate was calculated by averaging the deflection at the four nodal points at which the forces were applied and which are indicated in Figure 7. It was considered that this average would be more representative of the actual displacement, because of the relatively coarse finite-element grid. The results of these calculations are shown in Figure 8, where they are compared with the experimental results and the results calculated for the bottom plate by the hydrodynamic code HEMP. The experimental results are available only for the bottom part of the plate, which is opposite from the loaded surface. The results from the present analysis are shown for both the top and the bottom surfaces. Considering the coarsness of the trid, it is amazing that good agreement has been obtained between experimental and numerical results

## IV CONCLUSIONS

The method of analysis developed in this investigation have been successfully programmed for numerical calculations. Two different numerical integration of the dynamic equations. The relative merits of these programs can be stated at this time in terms of comparing application to relatively simple problems. In these applications it was found that the finite-difference method was the faster method although the disadvantage of this approach is that it is much more prone to numerical round off errors depending on the choice of the time interval size. The predictor-corrector method has a variable time interval which is automatically reduced according to some specified error limit.

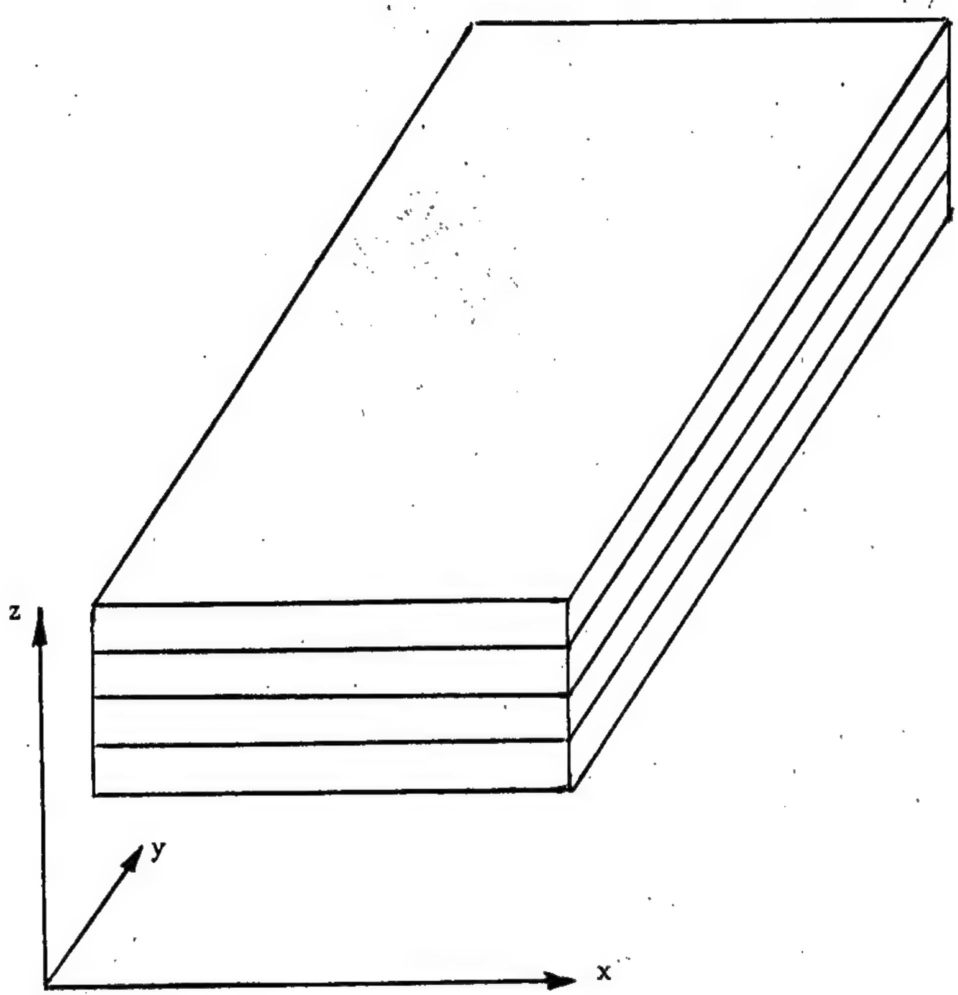


Figure 1. Coordinates used in the Analysis of Laminated Plate.

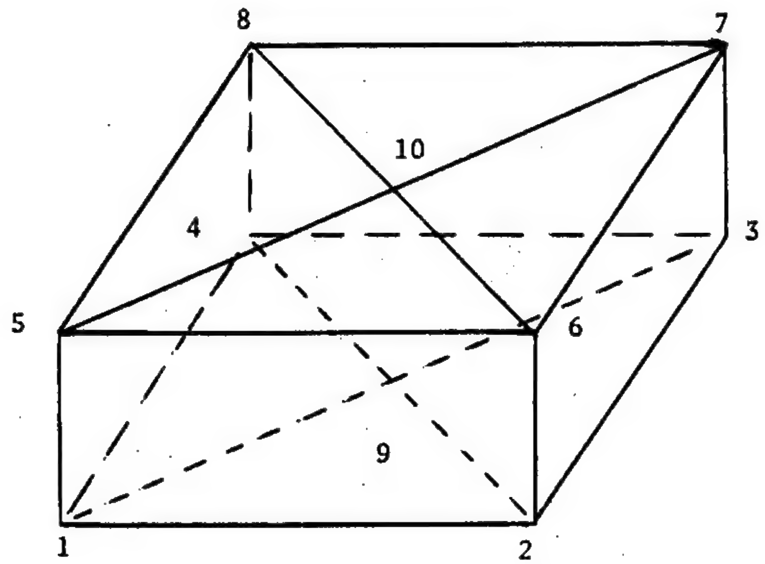


Figure 2. Nodal Numbering System
for Quadrilateral Element.

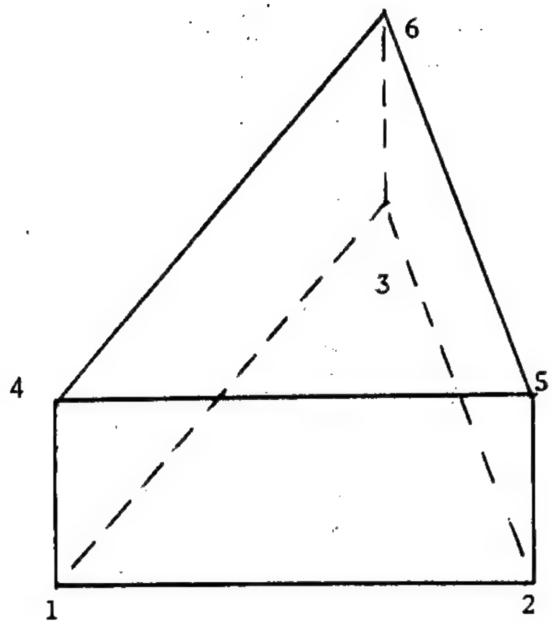


Figure 3. Nodal Numbering System for a Triangular Element.

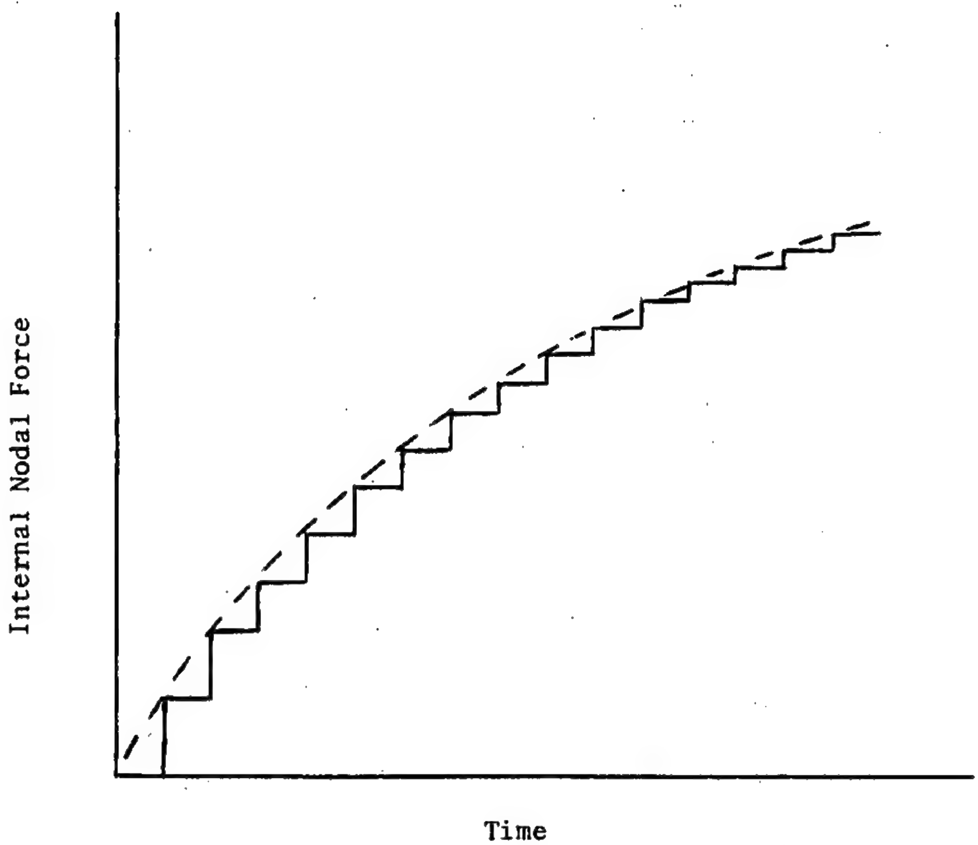


Figure 4. Step-Wise Representation of Internal Nodal Forces in the Finite-Difference Integration Method

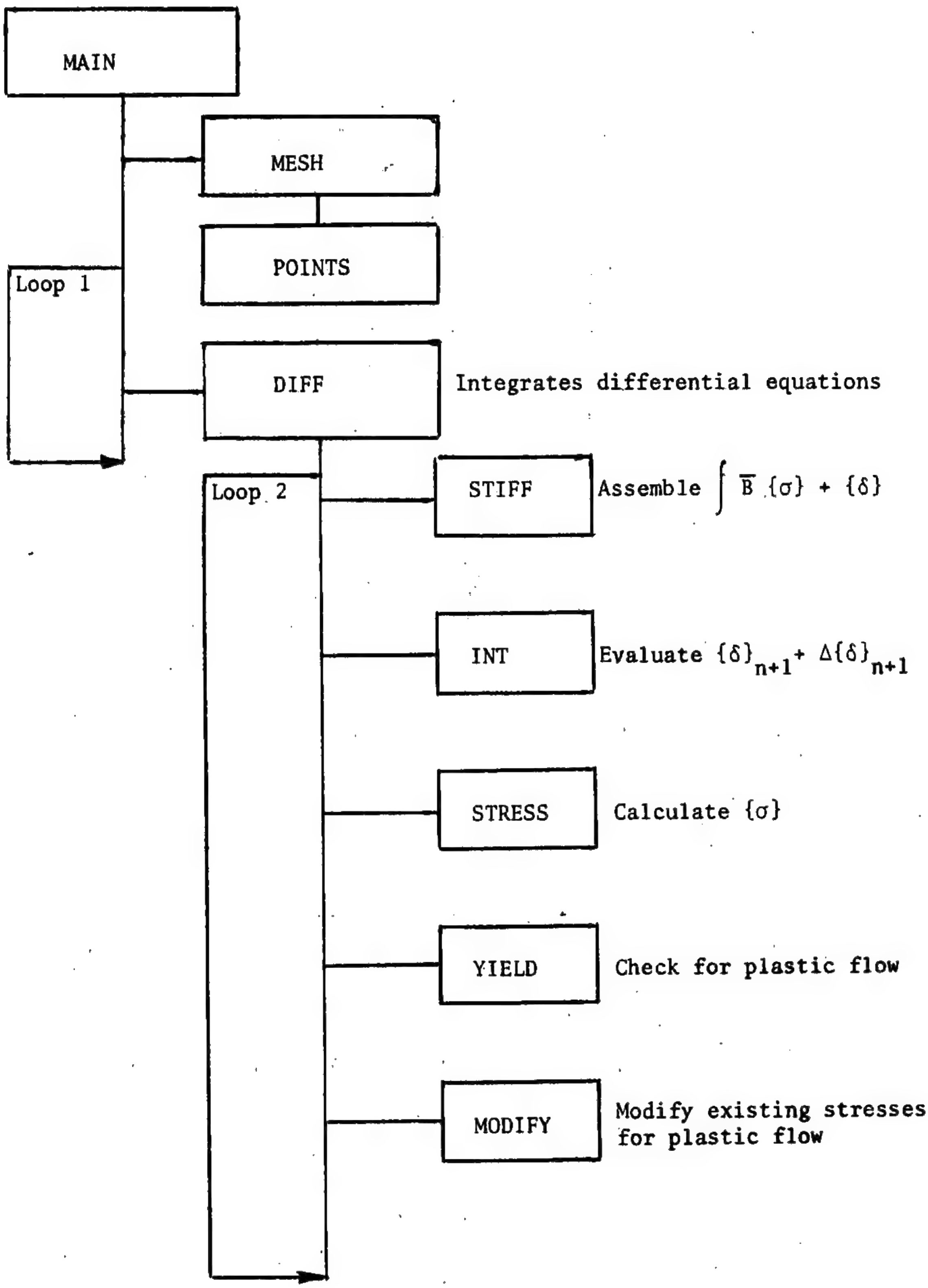


Figure 5. Computer Flow Chart for Finite Difference Method

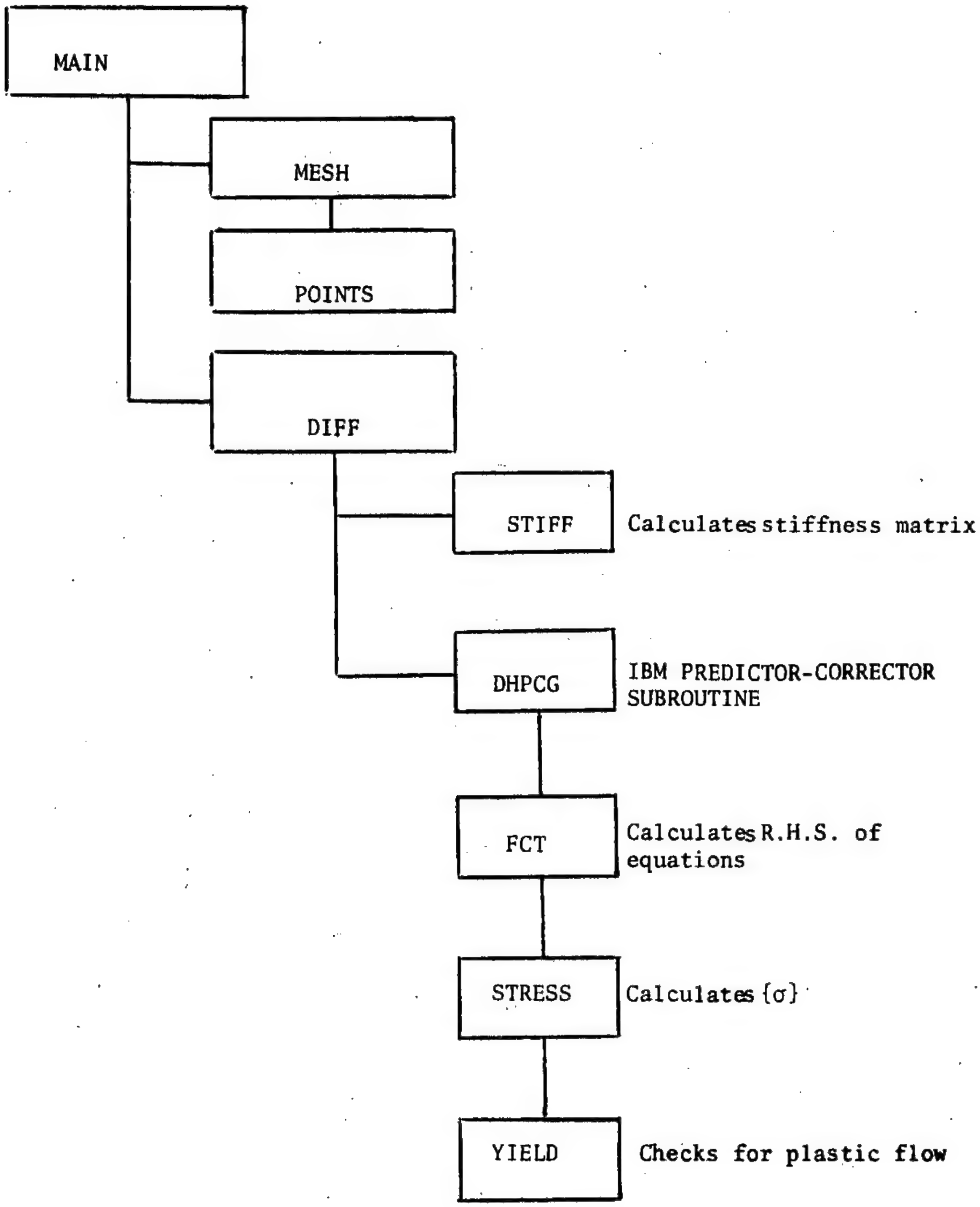


Figure 6. Computer Flow Chart for Predictor-Corrector Method

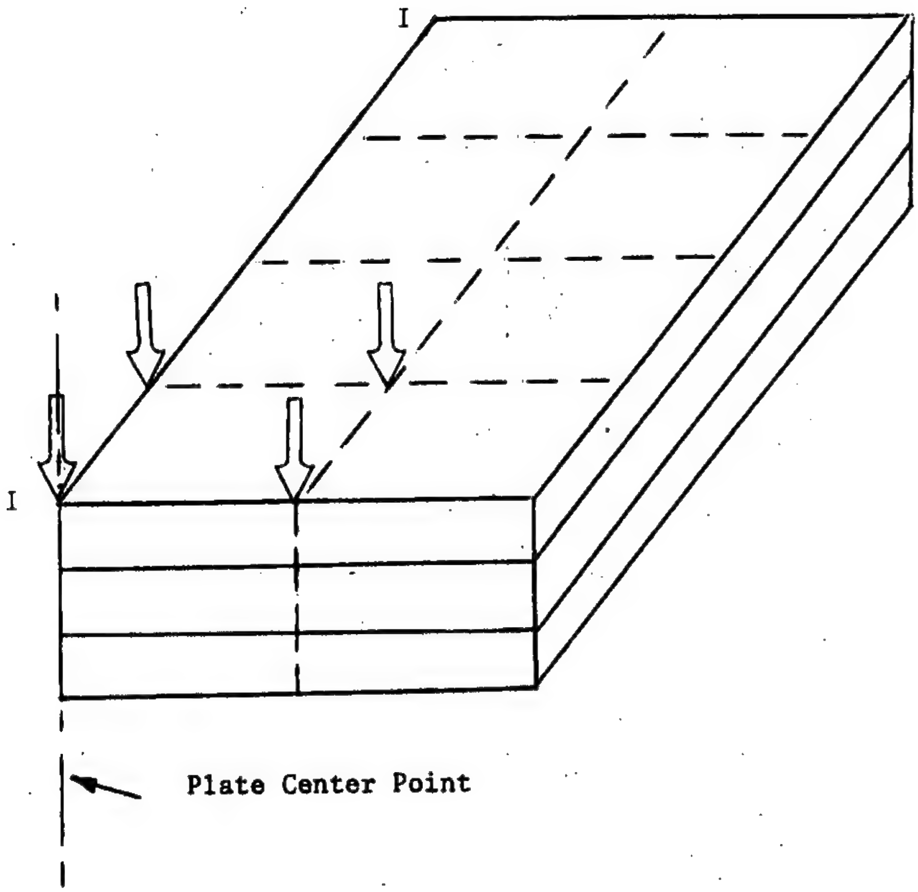


Figure 7. Finite Element Grid for Laminated Plate
Used in Numerical Example.

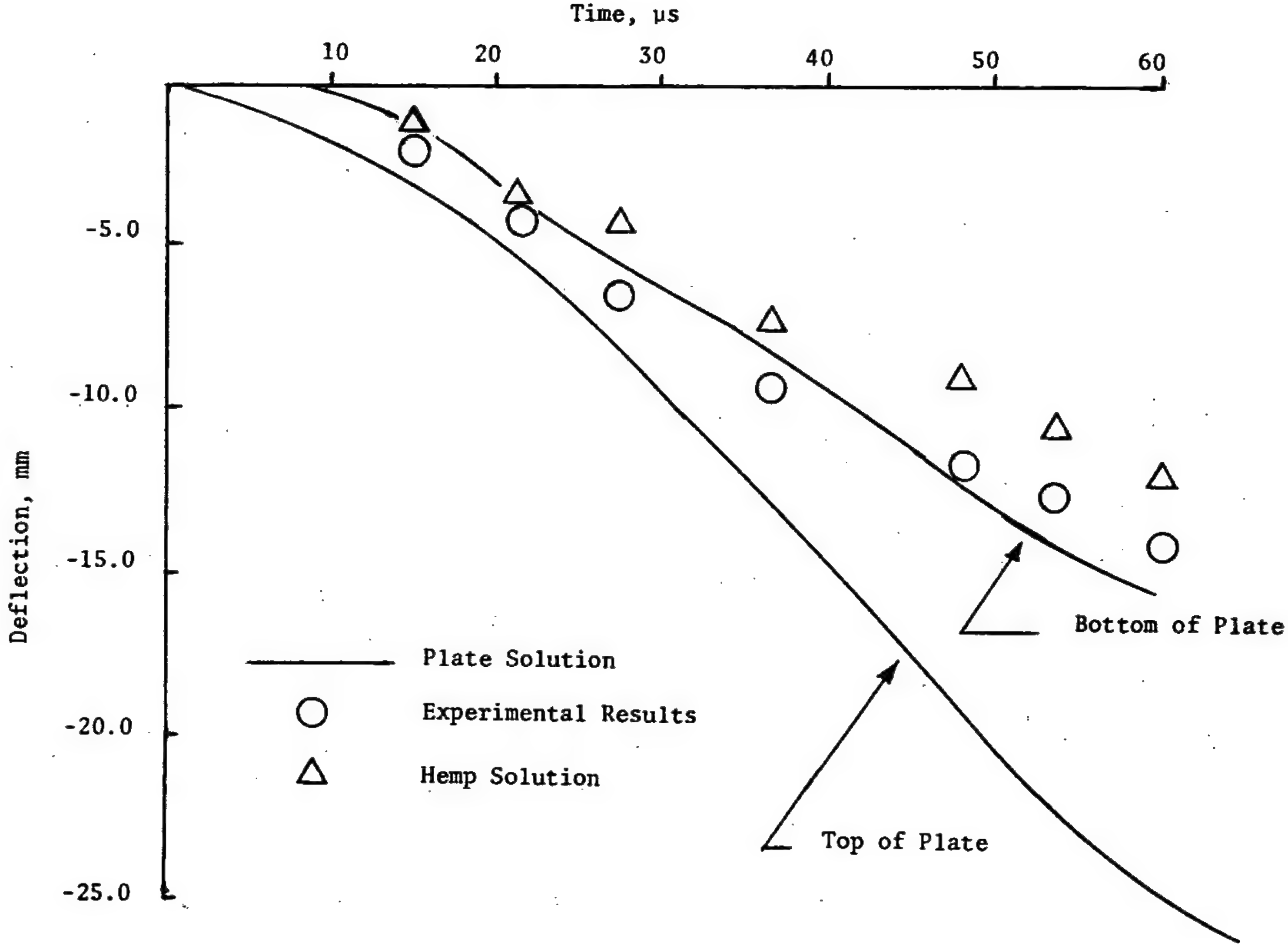


Figure 8. Comparison of Numerical and Experimental Results.

## APPENDIX A

## DERIVATION OF DIFFERENCE EQUATION OF MOTION

This Appendix shows the derivation of Equation (34). Consider Equations (33)

$$[M]\ \{\ddot{\Delta}\}_{n+1} = \{F_I\}_{n+1} + \{F_E\}_{n+1} \tag{A.1}$$

$$[M]\ \{\ddot{\Delta}\}_{n} = \{F_I\}_{n} + \{F_E\}_{n} \tag{A.2}$$

$$[M]\ \{\ddot{\Delta}\}_{n-1} = \{F_I\}_{n-1} + \{F_E\}_{n-1} \tag{A.3}$$

Applying the kinetic relations, Equations (31) and (32) to the displacement vector $\{\Delta\}$ gives for $t = (n+1)h$

$$\{\dot{\Delta}\}_{n+1} = \{\dot{\Delta}\}_n + \frac{h}{2}\ (\{\ddot{\Delta}\}_n + \{\ddot{\Delta}\}_{n+1}) \tag{A.4}$$

and

$$\{\Delta\}_{n+1} = \{\Delta\}_n + h\{\dot{\Delta}\}_n + (\frac{1}{2} - \beta)\ h^2\{\ddot{\Delta}\}_n$$

$$+ \beta h^2\{\ddot{\Delta}\}_{n+1} \tag{A.5}$$

Similar expressions are obtained for $t = nh$ by using Equations (29) and (30)

$$\{\dot{\Delta}\}_n = \{\dot{\Delta}\}_{n-1} + \frac{h}{2}\ (\{\ddot{\Delta}\}_{n-1} + \{\ddot{\Delta}\}_n) \tag{A.6}$$

and

$$\{\Delta\}_n = \{\Delta\}_{n-1} + h\{\dot{\Delta}\}_{n-1} + (\frac{1}{2} - \beta)h^2\{\ddot{\Delta}\}_{n-1}$$

$$+ \beta h^2\{\ddot{\Delta}\}_n \tag{A.7}$$

Multiplying Equation (A.2) by $2(\frac{1}{2} - \beta)h^2$ gives

$$2(\frac{1}{2} - \beta)h^2[M]\{\ddot{\Delta}\}_n = 2(\frac{1}{2} - \beta)h^2\ \{F_I\}_n + 2(\frac{1}{2} - \beta)h^2\ \{F_E\}_n \tag{A.8}$$

Multiplying Equations (A.1) and (A.3) by $\beta h^2$ results in

$$\beta h^2[M]\{\ddot{\Delta}\}_{n+1} = \beta h^2\{F_I\}_{n+1} + \beta h^2\{F_E\}_{n+1} \tag{A.9}$$

and

$$\beta h^2[M]\{\ddot{\Delta}\}_{n-1} = \beta h^2\ \{F_I\}_{n-1} + \beta h^2\{F_E\}_{n-1} \tag{A.10}$$

Adding Equations (A.8), (A.9), and (A.10) gives

$$\begin{aligned} \beta h^2[M]\{\ddot{\Delta}\}_{n+1}+2(\tfrac{1}{2} - \beta)h^2[M]\{\ddot{\Delta}\}_n + \beta h^2[M]\{\ddot{\Delta}\}_{n-1} \\ = \beta h^2\{F_I\}_{n+1} + 2(\tfrac{1}{2} - \beta)h^2\{F_I\}_n + \beta h^2\{F_I\}_{n-1} \\ + \beta h^2\{F_E\}_{n+1} + 2(\tfrac{1}{2} - \beta)h^2\{F_E\}_n + \beta h^2\{F_E\}_{n-1} \end{aligned} \tag{A.11}$$

or

$$\begin{aligned} [M]\{h^2[\beta\{\ddot{\Delta}\}_{n+1} + 2(\tfrac{1}{2} - \beta)\{\ddot{\Delta}\}_n + \beta\{\ddot{\Delta}\}_{n-1}]\} \\ = \beta h^2[\{F_I\}_{n+1} + (\tfrac{1}{\beta} - 2)\{F_I\}_n + \{F_I\}_{n-1}] \\ + \beta h^2[\{F_E\}_{n+1} + (\tfrac{1}{\beta} - 2)\{F_E\}_n + \{F_E\}_{n-1}] \end{aligned} \tag{A.12}$$

Consider the bracketed term in Equation (A.12), and let it be denoted by "Term 1," then

$$\text{Term 1} = \beta h^2\{\ddot{\Delta}\}_{n+1} + 2(\tfrac{1}{2} - \beta)h^2\{\ddot{\Delta}\}_n + \beta h^2\{\ddot{\Delta}\}_{n-1}$$

From Equation (A.5)

$$h^2\beta\{\ddot{\Delta}\}_{n+1} = \{\Delta\}_{n+1} - \{\Delta\}_n - h\{\dot{\Delta}\}_n - (\tfrac{1}{2} - \beta)h^2\{\ddot{\Delta}\}_n \tag{A.13}$$

Substituting Equation (A.13) in Term 1 results in

$$\begin{aligned} \text{Term 1} = \{\Delta\}_{n+1} - \{\Delta\}_n - h\{\dot{\Delta}\}_n - (\tfrac{1}{2} - \beta)h^2\{\ddot{\Delta}\}_n \\ + 2(\tfrac{1}{2} - \beta)\{\ddot{\Delta}\}_n + \beta\{\ddot{\Delta}\}_{n-1} \end{aligned}$$

$$= \{\Delta\}_{n+1} - \{\Delta\}_n - h\{\dot{\Delta}\}_n + (\frac{1}{2} - \beta)h^2\{\ddot{\Delta}\}_n$$

$$+ \beta h^2\{\ddot{\Delta}\}_{n-1}$$

$$= \{\Delta\}_{n+1} - \{\Delta\}_n - h\{\dot{\Delta}\}_n + \frac{h^2}{2}\{\ddot{\Delta}\}_n$$

$$- \beta h^2\{\ddot{\Delta}\}_n + \beta h^2\{\ddot{\Delta}\}_{n-1} + \{\ddot{\Delta}\}_{n-1}\frac{h^2}{2} - \{\ddot{\Delta}\}_{n-1}\frac{h^2}{2}$$

$$= \{\Delta\}_{n+1} - \{\Delta\}_n - h\{\dot{\Delta}\}_n + \frac{h^2}{2}(\{\ddot{\Delta}\}_n + \{\ddot{\Delta}\}_{n-1})$$

$$+ (- \beta h^2\{\ddot{\Delta}\}_n - h^2(\frac{1}{2} - \beta)\{\ddot{\Delta}\}_{n-1}) \tag{A.14}$$

From Equation (A.6)

$$\frac{h^2}{2}(\{\ddot{\Delta}\}_n + \{\ddot{\Delta}\}_{n-1}) = h(\{\dot{\Delta}\}_n - \{\dot{\Delta}\}_{n-1}) \tag{A.15}$$

From Equation (A.7)

$$- \beta h^2\{\ddot{\Delta}\}_n - h^2(\frac{1}{2} - \beta)\{\ddot{\Delta}\}_{n-1} = \{\Delta\}_{n-1} + h\{\dot{\Delta}\}_{n-1} - \{\Delta\}_n \tag{A.16}$$

Substituting Equation (A.15) and (A.16) in Equation (A.14) gives

$$\text{Term 1} = \{\Delta\}_{n+1} - \{\Delta\}_n - h\{\dot{\Delta}\}_n + h\{\dot{\Delta}\}_n - h\{\dot{\Delta}\}_{n-1} + \{\Delta\}_{n-1} + h\{\dot{\Delta}\}_{n-1} - \{\Delta\}_n$$

and after cancelling terms,

$$\text{Term 1} = \{\Delta\}_{n+1} - 2\{\Delta\}_n + \{\Delta\}_{n-1} \tag{A.17}$$

Therefore, upon substituting Equation (A.17), Equation (A.12) may be written as

$$[M]\ \{\{\Delta\}_{n+1} - 2\{\Delta\}_n + \{\Delta\}_{n-1}\}$$

$$= \beta h^2[\{F_E\}_{n+1} + (\frac{1}{\beta} - 2)\ \{F_E\}_n + \{F_E\}_{n-1}$$

$$+ \{F_I\}_{n+1} + (\frac{1}{\beta} - 2)\{F_I\}_n + \{F_I\}_{n-1}] \tag{A.18}$$

Rearranging Equation (A.18) gives

$$[M]\{\Delta\}_{n+1} - \beta h^2\{F_I\}_{n+1} = 2[M]\{\Delta\}_n$$

$$+ (1 - 2\beta)\, h^2\{F_I\}_n - [M]\{\Delta\}_{n-1}$$

$$+ \beta h^2\{F_I\}_{n-1}$$

$$+ h^2(\beta\{F_E\}_{n+1} + (1 - 2\beta)\{F_E\}_n + \beta\{F_E\}_{n-1})$$

which is the same as Equation (34).

# APPENDIX B

# COMPUTER PROGRAM INPUT

# CARD DESCRIPTION

TITLE CARD

Format (20A4) Title (Title for particular case)

CONTROL CARD

Format (4I5)

| | | |
|---|---|---|
| Columns | 1-5 | NUMMAT (Number of different materials; 6 maximum) |
| | 6-10 | NUMLA (Number of layers; 12 maximum) |
| | 11-15 | NLINC (Number of load increments with time; NLINC $\geq$ 1) |
| | 16-20 | IPLOT (Plot parameter, 1 if plot parameter, 1 if plot required) |

PRINT CARD 1-5 NPRINT (Number of intervals between printing)

INTEGRATION PARAMETER CARD

Format (F10.5, E15.7, I 10)

| | | |
|---|---|---|
| Columns | 1-10 | BET ($\beta$, acceleration parameter or Newmark's parameter, $0 \leq \beta \leq .25$) |
| | 11-25 | H (Time-step size) |
| | 26-35 | LINC (Magnitude of load increment) |

MESH GENERATION CONTROL CARD

Format (5I5)

| | | |
|---|---|---|
| Columns | 1-5 | MAXI (Maximum value of I in mesh; 25 maximum) |
| | 6-10 | MAXJ (Maximum value of J in mesh; 100 maximum) |
| | 11-15 | NSEG (Number of line segment cards) |
| | 16-20 | NBC (Number of boundary condition cards) |
| | 21-25 | NMTL (Number of material block cards) |

## LINE SEGMENT CARDS

The order of line segment cards is immaterial, except when plots are requested; in this case, the line segment cards must define the perimeter of solid continuously. The order of line segment cards defining internal straight lines is always irrelevant.

Format (3(2I3, 2F8.3), I5)

| | | |
|---|---|---|
| Columns | 1-3 | I coordinate of 1st point |
| | 4-6 | J coordinate of 1st point |
| | 7-14 | R coordinate of 1st point |
| | 15-22 | Z coordinate of 1st point |
| | 23-25 | I coordinate of 2nd point |
| | 26-28 | J coordinate of 2nd point |
| | 29-36 | R coordinate of 2nd point |
| | 37-44 | Z coordinate of 2nd point |
| | 45-47 | I coordinate of 3rd point |
| | 48-50 | J coordinate of 3rd point |
| | 51-58 | R coordinate of 3rd point |
| | 59-66 | Z coordinate of 3rd point |
| | 67-71 | Line segment type parameter |

If the number in column 71 is

| | |
|---|---|
| 0 | Point (input only 1st point). |
| 1 | straight line (input only 1st and 2nd points). |
| 2 | straight line as an internal diagonal (input only 1st and 2nd points). |
| 3 | circular arc specified by 1st and 3rd points at the ends of the arc and 2nd points at the mid-point of the arc. |
| 4 | circular arc specified by 1st and 2nd points at the ends of the arc with the coordinates of the center of the arc given as the 3rd point (delete I and J for 3rd point). |
| 5 | straight line as a boundary diagonal for which I of 1st point is minimum for its row and/or I or 2nd point is minimum for its row (input only 1st and 2nd points). |
| 6 | straight line as a boundary diagonal for which I of 1st point and/or 2nd point is maximum for its row (input only 1st and 2nd points). |

Note: In specifying a circular arc, the points are ordered such that a counter-clockwise direction about the center is obtained upon moving along the boundary.

## BOUNDARY CONDITION CARDS

Each card assigns a boundary condition code to a block of successive nodal points starting with N1 and ending with N2, inclusive.

Format (2I5, I10)

| | | |
|---|---|---|
| Columns | 1-5 | Starting node number N1 |
| | 6-10 | Ending node number N2 |
| | 11-20 | Boundary condition code |

If the number in columns 11-20 is;

| | |
|---|---|
| 0 | node is not restrained (program assigns automatically) |
| 1 | node is restrained in x direction |
| 2 | node is restrained in y direction |
| 3 | node is restrained in z direction |
| 4 | node is restrained in x and y directions |
| 5 | node is restrained in y and z directions |
| 6 | node is restrained in z and x directions |
| 7 | node is restrained in x, y, and z directions |

MATERIAL BLOCK ASSIGNMENT CARD

Each card assigns a material definition number to a block of elements defined by the I, J coordinates. One card for each layer.

Format (1I5, 3F10.0)

| | | |
|---|---|---|
| Columns | 1-5 | Material definition number (1 through 6) |
| | 6-15 | Material principal property inclination angle BETA in X-Y plane |
| | 16-25 | Material principal property inclination angle ALPHA in N-T plane |
| | 26-35 | Yield stress in this material layer |

PLOT TITLE CARD*

Format (20A4)

Columns 1-80 Title (Title printed under each plot)

PLOT GENERATION INFORMATION CARD*

Format (2F10.0)

| | | |
|---|---|---|
| Columns | 1-10 | RMAX (Maximum x coordinate of mesh) |
| | 11-20 | ZMAX (Maximum y coordinate of mesh) |

Note: Use only if IPLOT = 1 (plot required)

## MATERIAL PROPERTY INFORMATION CARDS

The following group of cards must be specified for each material (maximum of 6).

a. MATERIAL IDENTIFICATION CARD

Format (I15, F10.0)

| | | |
|---|---|---|
| Columns | 1-5 | Material identification number |
| | 6-15 | Mass density of material (if required) |

b. MATERIAL PROPERTY CARDS

First Card

Format (6F10.0)

| | | |
|---|---|---|
| Columns | 1-10 | Modulus of elasticity, $E_N$ |
| | 11-20 | Modulus of elasticity, $E_S$ |
| | 21-30 | Modulus of elasticity, $E_T$ |
| | 31-40 | Poisson's ratio, $\nu_{NS}$ |
| | 41-50 | Poisson's ratio, $\nu_{NT}$ |
| | 51-60 | Poisson's ratio, $\nu_{ST}$ |

Second Card

Format (3F10.0

| | | |
|---|---|---|
| Columns | 1-10 | Shear Modulus, $G_{NS}$ |
| | 11-20 | Shear Modulus, $G_{ST}$ |
| | 21-30 | Shear Modulus, $G_{TN}$ |

LAYER THICKNESS CARD

Format (12F5.3)

| | | |
|---|---|---|
| Columns | 1-5 | TH(1) (Thickness of layer 1) |
| | 6-10 | TH(2) (Thickness of layer 2) |
| | 11-15 | TH(3) (Thickness of layer 3) |

etc. up to TH (NUMLA)

# APPENDIX C

## COMPUTER PROGRAM LISTING FOR THE FINITE DIFFERENCE PROGRAM

```
      PROGRAM ARMYJOB(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,
     1TAPE2,TAPE3)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     BRLESC FINITE ELEMENT STRESS ANALYSIS OF AXISYMMETRIC,
C     PLANE STRAIN, AND PLANE STRESS SOLIDS WITH ORTHOTROPIC,
C     TEMPERATURE-DEPENDENT MATERIAL PROPERTIES
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/NPR/NPRINT
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/FIRST/FO(600),DER(600)
      COMMON/DELT/XDEL(4,18)
      COMMON/DELTRI/DELTA(4,18)
      COMMON/PLYLD/SIGY(12),DEPS(6)
      DIMENSION TITLE(20)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     READ AND WRITE CONTROL INFORMATION
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
   50 READ(5,1000)TITLE,NUMMAT,NUMLA,NLINC,IPLOT
      READ(5,1004)NPRINT
      IF(EOF(5))920,99
   99 WRITE(6,2000)TITLE,NUMLA,NUMMAT,NLINC
      READ(5,1002) BET,H,LINC
      WRITE(6,2001)BET,H,LINC
      TIME=0.00
      NTIME=0
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     GENERATE FINITE ELEMENT MESH
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  100 CALL MESH
      IF (IPLOT.EQ.1) CALL MPLOT
      MPRINT=0
      DO 230 N=1,NUMNP
      IF(MPRINT.NE.0) GO TO 220
      WRITE(6,2003)
      MPRINT=59
  220 MPRINT=MPRINT-1
  230 WRITE(6,2004) N,X(N),Y(N)
      IT=NUMNP*(NUMLA+1)*3
  440 MPRINT=0
      DO 460 N=1,NUMEL
      IF(MPRINT.NE.0) GO TO 450
      WRITE(6,2008)
      MPRINT=59
  450 MPRINT=MPRINT-1
      II=IX(N,1)
```

```
      JJ=IX(N,2)
      KK=IX(N,3)
      LL=IX(N,4)
  460 WRITE(6,2009) N,(IX(N,I),I=1,4)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     READ AND WRITE MATERIAL PROPERTIES
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  500 CONTINUE
      DO 510 M=1,NUMMAT
      READ(5,1004)  MTYPE,(RO(MTYPE))
      WRITE(6,2010) MTYPE,    RO(MTYPE)
      READ(5,1005)(E(J,MTYPE),J=1,9)
      WRITE(6,2011)(E(J,MTYPE),J=1,9)
  510 CONTINUE
      READ(5,1006)(TH(I),I=1,NUMLA)
      WRITE(6,1007)
 1007 FORMAT(" THICKNESSES")
      WRITE(6,1006)(TH(I),I=1,NUMLA)
      WRITE(6,1008)
 1008 FORMAT(" YIELD STRESSES")
      WRITE(6,1009)(SIGY(I),I=1,NUMLA)
 1009 FORMAT(" ",5(2X,E15.7))
      DO 800 I=1,NUMLA
      DO 800 J=1,NUMEL
      DO 800 K=1,6
      SIGMA(I,J,K)=0.00
  800 CONTINUE
      CALL INIT
      DO 900 NL=1,NLINC
      IF(NL.GT.1) GO TO 721
      DO 720 N=1,NUMLA
      ALPHA(N)=ALPHA(N)/57.295780
  720 BETA(N)=BETA(N)/57.295780
  721 CONTINUE
C
C     FORM STIFFNESS MATRIX
C
      DO 850 I=1,4
      DO 850 J=1,18
      DELTA(I,J)=0.00
  850 CONTINUE
      CALL DIFF
  900 CONTINUE
  910 GO TO 50
 1000 FORMAT(20A4/6I5,F5.0,5I5)
 1001 FORMAT(3F10.0)
 1002 FORMAT(E10.5,E15.7,I10)
 1004 FORMAT (I5,F10.0)
 1005 FORMAT(6F10.0)
 1006 FORMAT (12F5.3)
 2000 FORMAT (2H1 ,20A4/
     1 33H0  NUMBER OF LAYERS---------------I4/
     2 33H0  NUMBER OF MATERIALS------------I4/
     3 33H0  NUMBER OF LOAD INCREMENTS------I4/)
 2001 FORMAT(41H0  ACCELERATION PARAMETER, BETA-----------E10.5/
     148H0  TIME-STEP SIZE,H----------------------------E15.7/
     251H0  LOAD INCREMENT, LINC----------------------------I10/)
```

```
      2003 FORMAT (35H1    N        X           Y                )
      2004 FORMAT (I5,2F10.4)
      2008 FORMAT (51H1   EL     I     J    K    L         ANGLE BETA     ANGLE ALPHA)
      2009 FORMAT (I5,4I4,2F13.3)
      2010 FORMAT (1H1,"MATERIAL IDENTIFICATION NUMBER =",I2/
          21H ,"MASS DENSITY =",E15.7)
      2011 FORMAT (
          11H ,"MODULUS OF ELASTICITY-EN =",E15.7/
          21H ,"MODULUS OF ELASTICITY-ES =",E15.7/
          31H ,"MODULUS OF ELASTICITY-ET =",E15.7/
          41H ,"POISSON RATIO-NUNS =",E15.7/
          51H ,"POISSON RATIO-NUNT =",E15.7/
          61H ,"POISSON RATIO-NUST =",E15.7/
          71H ,"SHEAR MODULJS-GNS =",E15.7/
          81H ,"SHEAR MODULUS-GST =",E15.7/
          91H ,"SHEAR MODULUS-GTN =",E15.7/)
      2016 FORMAT (26H   THE SYSTEM CONVERGED IN I2,11H ITERATIONS)
      2017 FORMAT (33H   THE SYSTEM DID NOT CONVERGE IN I2,11H ITERATIONS)
       920 STOP
           END
```

```
ERITY  DETAILS     DIAGNOSIS OF PROBLEM

I                     CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY BE
I                     CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY BE
I                     CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY BE
I                     CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY BE
```

```
OLIC REFERENCE MAP (R=1)

S
JOB

   SN   TYPE              RELOCATION
        REAL         ARRAY     MASS               14  ALPHA      REAL         ARRAY     ELD
        REAL         ARRAY     MASS                0  BET        REAL                   BAS
        REAL         ARRAY     ELDATA              0  BSUM       REAL                   SIG
        REAL         ARRAY     RESULT           2260  CM         REAL         ARRAY     MAS
        REAL         ARRAY     RESULT            620  CODE       INTEGER      ARRAY     NPD
        REAL         ARRAY     ARG                 0  D          REAL         ARRAY     RES
        REAL         ARRAY     DISP             1130  DELN       REAL         ARRAY     DIS
1       REAL         ARRAY     DISP                0  DELTA      REAL         ARRAY     DEL
        REAL         ARRAY     PLYLD            1130  DER        REAL         ARRAY     FIR
        REAL         ARRAY     SIGM               14  E          REAL         ARRAY     MAT
        REAL         ARRAY     MATP             7035  F          REAL         ARRAY     SIG
        REAL         ARRAY     FIRST            3410  GNM1       REAL         ARRAY     DIS
        REAL         ARRAY     DISP                1  H          REAL                   BAS
        INTEGER                                    4  IFLAG      INTEGER                BAS
        INTEGER                                  144  IMAX       INTEGER      ARRAY     TD
        INTEGER      ARRAY     TD              13070  IPLOT      INTEGER
```

```
      SUBROUTINE ANGLE (R,Z,RC,ZC,ANG)
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(300)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     FIND ANGLE OF INCLINATION BETWEEN 0 AND 2*PI
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      PI=3.1415927
      D1=(Z-ZC)
      D2=(R-RC)
      IF(ABS(R-RC).GT.1.E-8) GO TO 100
      ANG=PI/2.
      IF(D1.GT.1.E-8) RETURN
      ANG=-ANG
      RETURN
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     ALLOW CIRCLE TO CROSS AXIS
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  100 ANG=ATAN2(D1,D2)
      RETURN
      END
```

OLIC REFERENCE MAP (R=1)

```
S
E

  SN   TYPE               RELOCATION
      REAL        ARRAY     MASS             14   ALPHA        REAL         ARRAY        ELD
      REAL                   F.P.          1130   B            REAL         ARRAY        MAS
      REAL                  BASIC2            0   BETA         REAL         ARRAY        ELD
      REAL                  SIGM             44   C            REAL         ARRAY        RES
      REAL        ARRAY     MASS            110   CNS          REAL         ARRAY        RES
      INTEGER     ARRAY     NPDATA           62   CRZ          REAL         ARRAY        ARG
      REAL        ARRAY     RESULT         2260   DEL          REAL         ARRAY        DIS
      REAL        ARRAY     DISP              0   DELN1        REAL         ARRAY        DIS
      REAL        ARRAY     SIGM             33   D1           REAL
      REAL                                   14   E            REAL         ARRAY        MAT
      REAL        ARRAY     MATP           7035   F            REAL         ARRAY        SIG
      REAL        ARRAY     DISP           4540   GNM2         REAL         ARRAY        DIS
      REAL                  BASIC2            4   IFLAG        INTEGER                   BAS
      INTEGER     ARRAY     TD                0   IMIN         INTEGER      ARRAY        TD
      INTEGER               BASIC2           44   IX           INTEGER      ARRAY        ELD
```

```
      SUBROUTINE CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      DIMENSION AR(10, 40),AZ(10, 40)
      EQUIVALENCE (X(1),AR),(Y(1),AZ)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     FIND INTERSECTION OF LINE AND CIRCLE = NEW R AND Z
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ANG1=ANG1+DELPHI
      RR=SQRT((RSTRT-RC)**2+(ZSTRT-ZC)**2)
      AR(I,J)=RC+RR*COS(ANG1)
      AZ(I,J)=ZC+RR*SIN(ANG1)
      RETURN
      END
```

OLIC REFERENCE MAP (R=1)

S
LE

| SN | TYPE | | RELOCATION | | | | | |
|---|---|---|---|---|---|---|---|---|
| | REAL | ARRAY | MASS | 14 | ALPHA | REAL | ARRAY | ELD/ |
| | REAL | | F.P. | 0 | AR | REAL | ARRAY | NPD/ |
| | REAL | ARRAY | NPDATA | 1130 | B | REAL | ARRAY | MAS |
| | REAL | | BASIC2 | 0 | BETA | REAL | ARRAY | ELD/ |
| | REAL | | SIGM | 44 | C | REAL | ARRAY | RES |
| | REAL | ARRAY | MASS | 110 | CNS | REAL | ARRAY | RES |
| | INTEGER | ARRAY | NPDATA | 62 | CRZ | REAL | ARRAY | ARG |
| | REAL | ARRAY | RESULT | 2260 | DEL | REAL | ARRAY | DIS |
| | REAL | ARRAY | DISP | 0 | DELN1 | REAL | ARRAY | DIS |
| HI | REAL | | F.P. | 7027 | DSIG | REAL | ARRAY | SIG |
| | REAL | ARRAY | MATP | 170 | EE | REAL | ARRAY | MAT |
| | REAL | ARRAY | SIGM | 3410 | GNM1 | REAL | ARRAY | DIS |
| | REAL | ARRAY | DISP | 1 | H | REAL | | BAS |
| | INTEGER | | F.P. | 4 | IFLAG | INTEGER | | BAS |
| | INTEGER | ARRAY | TD | 0 | IMIN | INTEGER | ARRAY | TD |
| | INTEGER | | BASIC2 | 44 | IX | INTEGER | ARRAY | ELD/ |
| | INTEGER | | F.P. | 341 | JMAX | INTEGER | ARRAY | TD |
| | INTEGER | ARRAY | TD | 7 | LINC | INTEGER | | BAS |
| | INTEGER | | ARG | 1504 | MATRIL | INTEGER | ARRAY | ELD |
| | INTEGER | | TD | 373 | MAXJ | INTEGER | | TD |
| | INTEGER | | ARG | 375 | NBC | INTEGER | | TD |

```
      SUBROUTINE DIFF
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      TIME=TIME+H
      NTIME=NTIME+1
      CALL STIFF
      CALL INT
      CALL STRESS
      RETURN
      END
```

OLIC REFERENCE MAP (R=1)

S

| SN | TYPE | | RELOCATION | | | | | |
|---|---|---|---|---|---|---|---|---|
| | REAL | ARRAY | MASS | 14 | ALPHA | REAL | ARRAY | ELDA |
| | REAL | ARRAY | MASS | 0 | BET | REAL | | BASI |
| | REAL | ARRAY | ELDATA | 0 | BSUM | REAL | | SIGM |
| | REAL | ARRAY | RESULT | 2260 | CM | REAL | ARRAY | MASS |
| | REAL | ARRAY | RESULT | 620 | CODE | INTEGER | ARRAY | NPDA |
| | REAL | ARRAY | ARG | 0 | D | REAL | ARRAY | RESU |
| | REAL | ARRAY | DISP | 1130 | DELN | REAL | ARRAY | DISP |
| 1 | REAL | ARRAY | DISP | 7027 | DSIG | REAL | ARRAY | SIGM |
| | REAL | ARRAY | MATP | 170 | EE | REAL | ARRAY | MATP |
| | REAL | ARRAY | SIGM | 3410 | GNM1 | REAL | ARRAY | DISP |
| | REAL | ARRAY | DISP | 1 | H | REAL | | BASI |
| G | INTEGER | | BASIC2 | 144 | IMAX | INTEGER | ARRAY | TD |
| | INTEGER | ARRAY | TD | 5 | IT | INTEGER | | BASI |
| | INTEGER | ARRAY | ELDATA | 341 | JMAX | INTEGER | ARRAY | TD |
| | INTEGER | ARRAY | TD | 7 | LINC | INTEGER | | BASI |
| | INTEGER | | ARG | 1504 | MATRIL | INTEGER | ARRAY | ELDA |
| | INTEGER | | TD | 373 | MAXJ | INTEGER | | TD |
| | INTEGER | | ARG | 375 | NBC | INTEGER | | TD |
| | INTEGER | | BASIC2 | 374 | NMTL | INTEGER | | TD |
| M | INTEGER | ARRAY | NPDATA | 6 | NTIME | INTEGER | | BASI |
| L | INTEGER | | BASIC | 3 | NUMLA | INTEGER | | BASI |
| P | INTEGER | | BASIC | 0 | RO | REAL | ARRAY | MATP |
| | REAL | ARRAY | ARG | 140 | SIG | REAL | ARRAY | ARG |
| A | REAL | ARRAY | SIGM | 30 | TH | REAL | ARRAY | ELDA |
| | REAL | | BASIC2 | 1 | TSUM | REAL | ARRAY | SIGM |

```
      SUBROUTINE DISFOR
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/FOR/FF(5),FC(5)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NUMB/NITER,HDAT,TO
      DIMENSION DIST(4)
      DIMENSION FF1(19)
      DATA (FF1(I),I=1,19)/1.2553135E-1,7.198626E-2,2.1038762E-2,
     1 8.6607670E-3,5.1174375E-3,3.5208787E-3,2.4258292E-3,
     1 1.7650479E-3,1.3472242E-3,1.0573703E-3,8.4967895E-4,
     1 7.0338441E-4,6.0297041E-4,5.3236837E-4,4.8614537E-4,
     1 4.3992239E-4,3.9369941E-4,3.4284337E-4,2.9198733E-4/
   11 NITER=19
      TNITER=36.0E-6
      HDAT=2.0E-6
      TO=0.0
      DO 10 I=1,4
      DIST(I)=SQRT(XXX(I)**2+YYY(I)**2)
   10 CONTINUE
      YBLAST=.2362*(TIME*1.0E+06+6.)
      IF(TIME-TNITER) 15,16,16
   15 CALL POINT(TIME,NUM,TN)
      DO 12 I=1,4
      FF(I)=-(FF1(NUM)+(FF1(NUM+1)-FF1(NUM))*(TIME-TN)/HDAT)*14.5E6
   12 CONTINUE
      GO TO 18
   16 DO 17 I=1,4
      FF(I)=-(FF1(NITER)+(FF1(NITER)-FF1(NITER-1))*(TIME-TNITER)/HDAT)
     1 *14.5E6
      IF(FF(I).LT.0.0) FF(I)=0.0
   17 CONTINUE
   18 CONTINUE
      DO 30 I=1,4
      IF(DIST(I).GT.YBLAST) FF(I)=0.0
   30 CONTINUE
      RETURN
      END
```

OLIC REFERENCE MAP (R=1)

S
OR

| | SN | TYPE | | RELOCATION | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | REAL | | BASIC2 | 62 | CRZ | REAL | ARRAY | ARG |
| | | REAL | ARRAY | | 5 | FC | REAL | ARRAY | FOR |
| | | REAL | ARRAY | FOR | 117 | FF1 | REAL | ARRAY | |
| | | REAL | | BASIC2 | 1 | HDAT | REAL | | NUMB |
| | | INTEGER | | | 4 | IFLAG | INTEGER | | BASI |
| | | INTEGER | | BASIC2 | 7 | LINC | INTEGER | | BASI |
| | | INTEGER | | ARG | 154 | N | INTEGER | | ARG |
| R | | INTEGER | | NUMB | 3 | NLAY | INTEGER | | BASI |

```
              SUBROUTINE INIT
              INTEGER CODE
              COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
              COMMON/MATP/RO(12),E(9,12),EE(9)
              COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
             1CRZ(6,6),XI(10),SIG(12),N,M
              COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
              COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
              COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
              COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
              COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
              COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
              COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
              COMMON/MASS/A(600),B(600),CM(8)
              COMMON/FIRST/FO(600),DER(600)
        C LET INITIAL DEFLECTION, VELOCITY, AND STRESS BE ZERO
        C LET INITIAL FORCE BE LINC
              DO 100 I=1,IT
              DELN(I)=0.00
              DER(I)=0.00
              GNM1(I)=0.00
              FO(I)=0.00
          100 CONTINUE
              RETURN
              END
```

```
OLIC REFERENCE MAP (R=1)

S

   SN  TYPE                RELOCATION
       REAL         ARRAY     MASS           14  ALPHA     REAL         ARRAY    ELDA
       REAL         ARRAY     MASS            0  BET       REAL                  BASI
       REAL         ARRAY     ELDATA          0  BSUM      REAL                  SIGM
       REAL         ARRAY     RESULT       2260  CM        REAL         ARRAY    MASS
       REAL         ARRAY     RESULT        620  CODE      INTEGER      ARRAY    NPDA
       REAL         ARRAY     ARG             0  D         REAL         ARRAY    RESU
       REAL         ARRAY     DISP         1130  DELN      REAL         ARRAY    DISP
 1     REAL         ARRAY     DISP         1130  DER       REAL         ARRAY    FIRS
       REAL         ARRAY     SIGM           14  E         REAL         ARRAY    MATP
       REAL         ARRAY     MATP         7035  F         REAL         ARRAY    SIGM
       REAL         ARRAY     FIRST        3410  GNM1      REAL         ARRAY    DISP
       REAL         ARRAY     DISP            1  H         REAL                  BASI
       INTEGER                                4  IFLAG     INTEGER               BASI
       INTEGER      ARRAY     TD              0  IMIN      INTEGER      ARRAY    TD
       INTEGER                BASIC2         44  IX        INTEGER      ARRAY    ELDA
       INTEGER      ARRAY     TD            310  JMIN      INTEGER      ARRAY    TD
       INTEGER                BASIC2        155  M         INTEGER               ARG
 IL    INTEGER      ARRAY     ELDATA        372  MAXI      INTEGER               TD
       INTEGER                TD            154  N         INTEGER               ARG
       INTEGER                TD              3  NLAY      INTEGER               BASI
       INTEGER                TD           1130  NPNUM     INTEGER      ARRAY    NPDA
```

```
      SUBROUTINE INT
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/FIRST/FD(600),DER(600)
      COMMON/MASS1/XMINV(600)
      DIMENSION NCOD(3)
      BH=BET *H**2
C H AND BETA ARE INPUT VARIABLES STORED IN COMMON
      DIF=H**2*(1-2.0*BET)
C OBTAIN INVERSE OF MASS MATRIX,XMINV
      IF(NTIME.GT.1)GO TO 201
      DO 200 I=1,IT
      XMINV(I)=1/A(I)
  200 CONTINUE
  201 CONTINUE
      NPT=(NUMLA+1)*NUMNP
      IF(NTIME.GT.1) GO TO 225
C STARTING PROCEDURE
      DIF1=0.50-BET
C DELN AND DER ARE INITIAL CONDITIONS ON DISPLACEMENT
C GNM1 IS INPUT VECTOR
      WRITE(6,999)
  999 FORMAT(" INITIAL FORCE VECTOR")
      WRITE(6,1000)(FD(I),I=1,IT)
 1000 FORMAT(9E12.6)
      DO 215 K=1,NPT
      ID=CODE(K)
      DO 205 J=1,3
  205 NCOD(J)=0
      IF(ID.EQ.1.OR.ID.EQ.4.OR.ID.EQ.7) NCOD(1)=1
      IF(ID.EQ.2.OR.ID.EQ.4.OR.ID.EQ.5.OR.ID.EQ.7) NCOD(2)=1
      IF(ID.EQ.3.OR.ID.EQ.5.OR.ID.EQ.6.OR.ID.EQ.7) NCOD(3)=1
      NP3=3*K
      DO 215 J=1,3
      I=NP3-3+J
      IF(NCOD(J).EQ.0) GO TO 210
      DEL(I)=0.00
      GO TO 215
  210 DEL(I)=DELN(I)+H*DER(I)
      DEL(I)=DEL(I)+XMINV(I)*(BH*B(I)-DIF1*H**2*GNM1(I))
      DEL(I)=DEL(I)+(DIF1*H**2*XMINV(I)*FD(I))
  215 CONTINUE
      GO TO 500
  225 DO 250 I=1,IT
      DELN1(I)=DELN(I)
      DELN(I)=DEL(I)
  250 CONTINUE
```

```
          DO 300 K=1,NPT
          ID=CODE(K)
          DO 270 J=1,3
  270     NCOD(J)=0
          IF(ID.EQ.1.OR.ID.EQ.4.OR.ID.EQ.7) NCOD(1)=1
          IF(ID.EQ.2.OR.ID.EQ.4.OR.ID.EQ.5.OR.ID.EQ.7) NCOD(2)=1
          IF(ID.EQ.3.OR.ID.EQ.5.OR.ID.EQ.6.OR.ID.EQ.7) NCOD(3)=1
          NP3=3*K
          DO 300 J=1,3
          I=NP3-3+J
          IF(NCOD(J).EQ.0) GO TO 275
          DEL(I)=0.00
          GO TO 300
  275     DEL(I)=DELN(I)*2-DELN1(I)
          DEL(I)=DEL(I)+XMINV(I)*(BH*B(I)+GNM1(I)*DIF+BH*GNM2(I))
  300     CONTINUE
  500     CONTINUE
          RETURN
          END
```

OLIC REFERENCE MAP (R=1)

S

| SN | TYPE | | RELOCATION | | | | | |
|---|---|---|---|---|---|---|---|---|
| | REAL | ARRAY | MASS | 14 | ALPHA | REAL | ARRAY | ELD |
| | REAL | ARRAY | MASS | 0 | BET | REAL | | BAS |
| | REAL | ARRAY | ELDATA | 244 | BH | REAL | | |
| | REAL | | SIGM | 2260 | CM | REAL | ARRAY | MASS |
| | INTEGER | ARRAY | NPDATA | 62 | CRZ | REAL | ARRAY | ARG |
| | REAL | ARRAY | DISP | 1130 | DELN | REAL | ARRAY | DISP |
| 1 | REAL | ARRAY | DISP | 1130 | DER | REAL | ARRAY | FIRS |
| | REAL | | | 250 | DIF1 | REAL | | |
| | REAL | ARRAY | SIGM | 14 | E | REAL | ARRAY | MATP |
| | REAL | ARRAY | MATP | 7035 | F | REAL | ARRAY | SIGM |
| | REAL | ARRAY | FIRST | 3410 | GNM1 | REAL | ARRAY | DISP |
| | REAL | ARRAY | DISP | 1 | H | REAL | | BAS |
| | INTEGER | | | 252 | ID | INTEGER | | |
| G | INTEGER | | BASIC2 | 144 | IMAX | INTEGER | ARRAY | TD |
| | INTEGER | ARRAY | TD | 5 | IT | INTEGER | | BAS |
| | INTEGER | ARRAY | ELDATA | 253 | J | INTEGER | | |
| | INTEGER | ARRAY | TD | 310 | JMIN | INTEGER | ARRAY | TD |
| | INTEGER | | | 7 | LINC | INTEGER | | BAS |
| | INTEGER | | ARG | 1504 | MATRIL | INTEGER | ARRAY | ELD |
| | INTEGER | | TD | 373 | MAXJ | INTEGER | | TD |
| | INTEGER | | ARG | 375 | NBC | INTEGER | | TD |
| | INTEGER | ARRAY | | 3 | NLAY | INTEGER | | BAS |
| | INTEGER | | TD | 1130 | NPNUM | INTEGER | ARRAY | NPD |
| | INTEGER | | | 254 | NP3 | INTEGER | | |
| E | INTEGER | | BASIC2 | 2 | NUMEL | INTEGER | | BAS |
| A | INTEGER | | BASIC | 1 | NUMNP | INTEGER | | BAS |
| | REAL | ARRAY | MATP | 24 | S | REAL | ARRAY | ARG |



1

```
      SUBROUTINE INTER
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      DIMENSION X(7),Y(7),QQ(9)
      DATA QQ/3*.1259391805448,3*.1323941527884,.225,
     1 .696140478028,.4104261923147
      X(7)=(XX(1)+XX(2)+XX(3))/3.0
      Y(7)=(YY(1)+YY(2)+YY(3))/3.0
      DO 100 I=1,3
      J=I+3
      X(I)=QQ(8)*XX(I)+(1.00-QQ(8))*X(7)
      X(J)=QQ(9)*XX(I)+(1.00-QQ(9))*X(7)
      Y(I)=QQ(8)*YY(I)+(1.00-QQ(8))*Y(7)
  100 Y(J)=QQ(9)*YY(I)+(1.00-QQ(9))*Y(7)
      DO 300 I=1,10
  300 XI(I)=0.00
      AREA=.5)*(XX(1)*(YY(2)-YY(3))+XX(2)*(YY(3)-YY(1))+XX(3)*(YY(1)
     1  -YY(2)))
      DO 400 I=1,7
      XI(1)=XI(1)+QQ(I)
      XI(2)=XI(2)+QQ(I)*X(I)
      XI(3)=XI(3)+QQ(I)*Y(I)
      XI(4)=XI(4)+QQ(I)*X(I)*Y(I)
      XI(5)=XI(5)+QQ(I)*X(I)**2
  400 XI(6)=XI(6)+QQ(I)*Y(I)**2
      DO 500 I=1,10
  500 XI(I)=XI(I)*AREA
      RETURN
      END
```

```
1BOLIC REFERENCE MAP (R=1)

ITS
TER

      SN   TYPE                 RELOCATION
           REAL       ARRAY     MASS          14   ALPHA    REAL     ARRAY    EL
:A         REAL                             1130   B        REAL     ARRAY    MA
           REAL                 BASIC2         0   BETA     REAL     ARRAY    EL
JM         REAL                 SIGM          44   C        REAL     ARRAY    RE
           REAL       ARRAY     MASS         110   CNS      REAL     ARRAY    RE
)E         INTEGER    *UNDEF                  62   CRZ      REAL     ARRAY    AR
           REAL       ARRAY     RESULT      2260   DEL      REAL     ARRAY    DI
```

```
      SUBROUTINE LOAD
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/FOR/FF(5),FC(5)
      DO 50 I=1,IT
      F(I)=0.00
   50 CONTINUE
C     NUMBER NODES OF THE RECTANGULAR ELEMENTS
      DO 20 N=1,NUMEL
      I1=IX(N,1)
      I2=IX(N,2)
      I3=IX(N,3)
      I4=IX(N,4)
C     DESIGNATE COORDINATES
      XXX(1)=X(I1)
      XXX(2)=X(I2)
      XXX(3)=X(I3)
      XXX(4)=X(I4)
      XXX(5)=1.0/4.0*(XXX(1)+XXX(2)+XXX(3)+XXX(4))
      YYY(1)=Y(I1)
      YYY(2)=Y(I2)
      YYY(3)=Y(I3)
      YYY(4)=Y(I4)
      YYY(5)=1.0/4.0*(YYY(1)+YYY(2)+YYY(3)+YYY(4))
      CALL DISFOR
      IF(FF(1).EQ.0.0.AND.FF(2).EQ.0.0.AND.FF(3).EQ.0.0.AND.FF(4).EQ.0.0
     1) GO TO 20
C     FIND FORCE AT CENTER OF ELEMENT
      FF(5)=1.0/4.0*(FF(1)+FF(2)+FF(3)+FF(4))
C     CALL LOADS OF TRIANGLES
      DO 9 I=1,5
      FC(I)=0.00
    9 CONTINUE
      CALL LOT(1,2)
      CALL LOT(2,3)
      CALL LOT(3,4)
      CALL LOT(4,1)
      DO 21 I=1,4
      FC(I)=FC(I)+FC(5)/4.0
   21 CONTINUE
C     CHANGE TO GLOBAL FORCES IN W-DIRECTION
      II1=NUMLA*NUMNP*3+I1*3
      II2=NUMLA*NUMNP*3+I2*3
      II3=NUMLA*NUMNP*3+I3*3
      II4=NUMLA*NUMNP*3+I4*3
      F(II1)=F(II1)+FC(1)
      F(II2)=F(II2)+FC(2)
      F(II3)=F(II3)+FC(3)
      F(II4)=F(II4)+FC(4)
   20 CONTINUE
      RETURN
```

```
      SUBROUTINE LOT(II,JJ)
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/FOR/FF(5),FC(5)
      DIMENSION DD1(3),DD2(3),DD3(3),DD(3,3),FE(3),FFF(3)
      DIMENSION AA(3),BB(3),CC(3)
C     DESIGNATE THE TRIANGULAR DISTRIBUTED FORCES
      FFF(1)=FF(II)
      FFF(2)=FF(JJ)
      FFF(3)=FF(5)
      XX(1)=XXX(II)
      XX(2)=XXX(JJ)
      XX(3)=XXX(5)
      YY(1)=YYY(II)
      YY(2)=YYY(JJ)
      YY(3)=YYY(5)
      AA(1)=XX(2)*YY(3)-XX(3)*YY(2)
      AA(2)=XX(3)*YY(1)-XX(1)*YY(3)
      AA(3)=XX(1)*YY(2)-XX(2)*YY(1)
      BB(1)=YY(2)-YY(3)
      BB(2)=YY(3)-YY(1)
      BB(3)=YY(1)-YY(2)
      CC(3)=XX(2)-XX(1)
      CC(2)=XX(1)-XX(3)
      CC(1)=XX(3)-XX(2)
C     INTEGRATE XX AND YY
      CALL INTER
      DO 12 I=1,3
      DD1(I)=AA(I)*XI(1)+BB(I)*XI(2)+CC(I)*XI(3)
      DD2(I)=AA(I)*XI(2)+BB(I)*XI(5)+CC(I)*XI(4)
      DD3(I)=AA(I)*XI(3)+BB(I)*XI(4)+CC(I)*XI(6)
   12 CONTINUE
      DO 18 I=1,3
      DO 18 J=1,3
      DD(I,J)=AA(I)*DD1(J)+BB(I)*DD2(J)+CC(I)*DD3(J)
   18 CONTINUE
C     CALCULATE EQUIVELENT CONCENTRATED FORCES
      AREA=.50*(XX(1)*(YY(2)-YY(3))+XX(2)*(YY(3)-YY(1))+XX(3)*(YY(1)
     1 -YY(2)))
      DO 99 I=1,3
      FE(I)=1.0/(4.0*AREA**2)*(DD(I,1)*FFF(1)+DD(I,2)*FFF(2)+DD(I,3)*FFF
     1(3))
   99 CONTINUE
      FC(II)=FC(II)+FE(1)
      FC(JJ)=FC(JJ)+FE(2)
      FC(5)=FC(5)+FE(3)
      RETURN
      END
```

```
      SUBROUTINE MESH
      INTEGER CODE
      DIMENSION AR(10, 40), AZ(10, 40),NCODE(10, 40)
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      EQUIVALENCE (X(1),AR),(Y(1),AZ),(IX(1,1),NCODE)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     MESH CONTROL INFORMATION
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      READ (5,1000) MAXI,MAXJ,NSEG,NBC,NMTL
      WRITE(6,2000) MAXI,MAXJ,NSEG,NBC,NMTL
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     INITIALIZE
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ISEG=-1
      PI=3.1415927
      DO 110 J=1,40
      DO 100 I=1,10
      NCODE(I,J)=0
      AR(I,J)=0.
      AZ(I,J)=0.
      JMAX(I)=0
  100 JMIN(I)=MAXI
      IMIN(J)=MAXJ
  110 IMAX(J)=0
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     LINE SEGMENT CARDS
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  150 ISEG=ISEG+1
  159 IF(ISEG.EQ.NSEG) GO TO 400
      READ(5,1001) I1,J1,R1,Z1,I2,J2,R2,Z2,I3,J3,R3,Z3,IPTION
      WRITE(6,2001)I1,J1,R1,Z1,I2,J2,R2,Z2,I3,J3,R3,Z3,IPTION
      IPTION=IPTION+1
      AR(I1,J1)=R1
      AZ(I1,J1)=Z1
      NCODE(I1,J1)=1
      CALL MNIMX(I1,J1)
      GO TO (150,200,200,300,300,200,200), IPTION
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     GENERATE STRAIGHT LINES ON BOUNDARY
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
200   DI= ABS(FLOAT(I2-I1))
      DJ= ABS(FLOAT(J2-J1))
      AR(I2,J2)=R2
      AZ(I2,J2)=Z2
      NCODE(I2,J2)=1
      CALL MNIMX(I2,J2)
```

```
      ISTRT=I1
      ISTP=I2
      JSTRT=J1
      JSTP=J2
      DIFF=MAX1(DI,DJ)
      ITER=DIFF-1.
      IINC=0
      JINC=0
      IF(I2.NE.I1) IINC=(I2-I1)/IABS(I2-I1)
      IF(J2.NE.J1) JINC=(J2-J1)/IABS(J2-J1)
      KAPPA=1
      IF(I2.NE.I1.AND.J2.NE.J1.AND.IPTION.NE.3) KAPPA=2
      IF(KAPPA.EQ.2) DIFF=2.*DIFF
      RINC=(R2-R1)/DIFF
      ZINC=(Z2-Z1)/DIFF
      WRITE(6,2002) DI,DJ,DIFF,RINC,ZINC,ITER,IINC,JINC,KAPPA
C
C     CHECK FOR INPUT ERROR
C
      IF(KAPPA.NE.2.OR.DI.EQ.DJ) GO TO 210
      WRITE(6,2003)
      GO TO 150
C
C     INTERPOLATE
C
  210 I=I1
      J=J1
      WRITE(6,2004)
      DO 230 M=1,ITER
      IF(ITER.EQ.0.AND.IPTION.EQ.2) GO TO 230
      IF(ITER.EQ.0.AND.IPTION.EQ.6) GO TO 230
      IF(ITER.EQ.0.AND.IPTION.EQ.7) GO TO 230
      IF(KAPPA.EQ.2) GO TO 220
      IOLD=I
      I=I+IINC
      JOLD=J
      J=J+JINC
      AR(I,J)=AR(IOLD,JOLD)+RINC
      AZ(I,J)=AZ(IOLD,JOLD)+ZINC
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      CALL MNIMX(I,J)
      NCODE(I,J)=1
      GO TO 230
  220 CONTINUE
      IF(I1.GT.I2.AND.IPTION.EQ.7) GO TO 221
      IF(I1.LT.I2.AND.IPTION.EQ.6) GO TO 221
      IOLD=I
      I=I+IINC
      AR(I,J)=AR(IOLD,J)+RINC
      AZ(I,J)=AZ(IOLD,J)+ZINC
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      NCODE(I,J)=1
      CALL MNIMX(I,J)
      JOLD=J
      J=J+JINC
      AR(I,J)=AR(I,JOLD)+RINC
      AZ(I,J)=AZ(I,JOLD)+ZINC
```

```
          NCODE(I,J)=1
          WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
          CALL MNIMX(I,J)
          GO TO 230
      221 JOLD=J
          J=J+JINC
          AR(I,J)=AR(I,JOLD)+RINC
          AZ(I,J)=AZ(I,JOLD)+ZINC
          NCODE(I,J)=1
          WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
          CALL MNIMX(I,J)
          IOLD=I
          I=I+IINC
          AR(I,J)=AR(IOLD,J)+RINC
          AZ(I,J)=AZ(IOLD,J)+ZINC
          NCODE(I,J)=1
          WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
          CALL MNIMX(I,J)
      230 CONTINUE
          IF(KAPPA.EQ.1) GO TO 150
          IF(I1.GT.I2.AND.IPTION.EQ.7) GO TO 231
          IF(I1.LT.I2.AND.IPTION.EQ.6) GO TO 231
          IOLD=I
          I=I+IINC
          AR(I,J)=AR(IOLD,J)+RINC
          AZ(I,J)=AZ(IOLD,J)+ZINC
          GO TO 232
      231 CONTINUE
          JOLD=J
          J=J+JINC
          AR(I,J)=AR(I,JOLD)+RINC
          AZ(I,J)=AZ(I,JOLD)+ZINC
      232 CONTINUE
          NCODE(I,J)=1
          WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
          CALL MNIMX(I,J)
          GO TO 150
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C         GENERATE CIRCULAR ARCS ON BOUNDARY
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      300 AR(I2,J2)=R2
          AZ(I2,J2)=Z2
          NCODE(I2,J2) = 1
          CALL MNIMX(I2,J2)
          IF(IPTION.EQ.5) GO TO 320
C
C         FIND CENTER OF CIRCLE
C
          AR(I3,J3)=R3
          AZ(I3,J3)=Z3
          NCODE(I3,J3)=1
          CALL MNIMX(I3,J3)
          SLAC=(Z2-Z1)/(R2-R1)
          SLBF=-1./SLAC
          SLCE=(Z3-Z2)/(R3-R2)
          SLDF=-1./SLCE
C
```

```
C       CHECK FOR INPUT ERROR
C
        IF(ABS(SLAC-SLCE).GT..001) GO TO 310
        WRITE(6,2006) R1,Z1,R2,Z2,R3,Z3,SLAC,SLCE
        GO TO 150
  310   R4=R1+(R2-R1)/2.
        Z4=Z1+(Z2-Z1)/2.
        R5=R2+(R3-R2)/2.
        Z5=Z2+(Z3-Z2)/2.
        BBF=Z4-SLBF*R4
        BDF=Z5-SLDF*R5
        RC=(BBF-BDF)/(SLDF-SLBF)
        ZC=SLBF*RC+BBF
        WRITE(6,2007) RC,ZC
        KAPPA=1
        GO TO 330
  320   KAPPA=2
        RC=R3
        ZC=Z3
  330   ISTRT=I1
        ISTP=I2
        JSTRT=J1
        JSTP=J2
        RSTRT=R1
        RSTP=R2
        ZSTRT=Z1
        ZSTP=Z2
  340   CALL ANGLE(RSTRT,ZSTRT,RC,ZC,ANG1)
        CALL ANGLE(RSTP,ZSTP,RC,ZC,ANG2)
        IF(ANG2.LE.ANG1)  ANG2=2.0*PI+ANG2
C
C       FIND ANGULAR INCREMENT
C
        DI= ABS(FLOAT(ISTP-ISTRT))
        DJ= ABS(FLOAT(JSTP-JSTRT))
        IINC=0
        JINC=0
        IF(ISTRT.NE.ISTP) IINC=(ISTP-ISTRT)/IABS(ISTP-ISTRT)
        IF(JSTRT.NE.JSTP) JINC=(JSTP-JSTRT)/IABS(JSTP-JSTRT)
        LAMDA=1
        IF(IINC.NE.0.AND.JINC.NE.0) LAMDA=2
        DIFF=MAX1(DI,DJ)
        ITER=DIFF-1.
        IF(LAMDA.EQ.2) DIFF=2.*DIFF
        DELPHI=(ANG2-ANG1)/DIFF
        WRITE(6,2008) ANG1,ANG2,DIFF,DELPHI
C
C       CHECK FOR INPUT ERROR
C
        IF(LAMDA.NE.2.OR.DI.EQ.DJ) GO TO 350
        WRITE(6,2003)
        GO TO 150
  350   IO=ISTRT
        JO=JSTRT
        WRITE(6,2004)
C
C       INTERPOLATE
```

```
C
          NPT=IABS(I2-I1)+IABS(J2-J1)-1
          DO 380 M=1,ITER
      359 IF(LAMDA.EQ.2) GO TO 360
          I=IO+IINC
          J=JO+JINC
          CALL MNIMX(I,J)
          NCODE(I,J)=1
          CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
          WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
          GO TO 370
      360 I=IO+IINC
          J=JO
          NCODE(I,J)=1
          CALL MNIMX(I,J)
          CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
          WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
          J=JO+JINC
          NCODE(I,J)=1
          CALL MNIMX(I,J)
          CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
          WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      370 IO=I
      380 JO=J
          IF(LAMDA.NE.2) GO TO 390
          I=IO+IINC
          NCODE(I,J)=1
          CALL MNIMX(I,J)
          CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
          WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      390 IF(KAPPA.EQ.2) GO TO 150
          ISTRT=I2
          ISTP=I3
          JSTRT=J2
          JSTP=J3
          RSTRT=R2
          RSTP=R3
          ZSTRT=Z2
          ZSTP=Z3
          KAPPA=2
      399 GO TO 340
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C         CALCULATE COORDINATES OF INTERIOR POINTS
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      400 IF(MAXJ.LE.2) GO TO 430
          J2=MAXJ-1
          DO 420 N=1,500
          RESID=0.
          DO 410 J=2,J2
          I1=IMIN(J)+1
          I2=IMAX(J)-1
          DO 410 I=I1,I2
          IF(NCODE(I,J).EQ.1) GO TO 410
          DR=(AR(I+1,J)+AR(I-1,J)+AR(I,J+1)+AR(I,J-1))/4.-AR(I,J)
          DZ=(AZ(I+1,J)+AZ(I-1,J)+AZ(I,J+1)+AZ(I,J-1))/4.-AZ(I,J)
          RESID=RESID+ABS(DR)+ABS(DZ)
          AR(I,J)=AR(I,J)+1.8*DR
```

```
          AZ(I,J)=AZ(I,J)+1.8*DZ
   410    CONTINUE
          IF(N.EQ.1) RES1=RESID
          IF(N.EQ.1.AND.RESID.EQ.0.)GO TO 430
          IF(RESID/RES1.LT.1.E-5) GO TO 430
   420    CONTINUE
   430    WRITE(6,2009) N
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
          CALL POINTS
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  1000    FORMAT (5I5)
  1001    FORMAT (3(2I3,2F8.3),I5)
  2000    FORMAT (30H1  MESH GENERATION INFORMATION//
         1 41H0   MAXIMUM VALUE OF I IN THE MESH---------I3/
         2 41H0   MAXIMUM VALUE OF J IN THE MESH---------I3/
         3 41H0   NUMBER OF LINE SEGMENT CARDS-----------I3/
         4 41H0   NUMBER OF BOUNDARY CONDITION CARDS-----I3/
         5 41H0   NUMBER OF MATERIAL BLOCK CARDS---------I3///)
  2001    FORMAT (//88H   INPUT    I1   J1      R1          Z1      I2   J2      R2        Z
         12      I3   J3      R3          Z3      IPTION/8X,3(2I4,2F8.4),I6)
  2002    FORMAT (5H  DI=F4.0,5H  DJ=F4.0,7H  DIFF=F4.0,7H  RINC=F8.3,7H  ZI
         1NC=F8.3,7H  ITER=I3,7H  IINC=I3,7H  JINC=I3,8H  KAPPA=I1)
  2003    FORMAT(1X,38H**BAD INPUT--THIS LINE IS NOT DIAGONAL)
  2004    FORMAT (30H     I       J          AR          AZ)
  2005    FORMAT (2I5,2F11.6)
  2006    FORMAT (51H ** BAD INPUT - THESE POINTS DO NOT DEFINE A CIRCLE,/,
         13X,6F12.4,10X,2E20.8)
  2007    FORMAT(19H  CENTER COORDINATE,(F11.6,1X,F11.6,1X))
  2008    FORMAT (7H  ANG1=F9.6,7H  ANG2=F9.6,7H  DIFF=F3.0,9H  DELPHI=F9.6)
  2009    FORMAT (//30H  COORDINATES CALCULATED AFTER I3,11H ITERATIONS)
          RETURN
          END
```

```
EVERITY  DETAILS    DIAGNOSIS OF PROBLEM

  I                    CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY B
  I                    CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY B
  I                    CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY B
```

```
MBOLIC REFERENCE MAP (R=1)

NTS
SH

       SN  TYPE          RELOCATION
           REAL     ARRAY     MASS           14  ALPHA      REAL          ARRAY     EL
G1         REAL                            1625  ANG2       REAL
           REAL     ARRAY     NPDATA        310  AZ         REAL          ARRAY     NP
           REAL     ARRAY     MASS         1614  BBF        REAL
F          REAL                               0  BET        REAL                    BA
```

```
      SUBROUTINE MNIMX(I,J)
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      IF(J.LT.JMIN(I)) JMIN(I)=J
      IF(J.GT.JMAX(I)) JMAX(I)=J
      IF(I.LT.IMIN(J)) IMIN(J)=I
      IF(I.GT.IMAX(J)) IMAX(J)=I
      RETURN
      END
```

MBOLIC REFERENCE MAP (R=1)

NTS
IMX

| SN | TYPE | | RELOCATION | | | | | |
|---|---|---|---|---|---|---|---|---|
| | REAL | ARRAY | MASS | 14 | ALPHA | REAL | ARRAY | EL |
| | REAL | ARRAY | MASS | 0 | BET | REAL | | BA |
| TA | REAL | ARRAY | ELDATA | 0 | BSUM | REAL | | SI |
| | REAL | ARRAY | RESULT | 2260 | CM | REAL | ARRAY | MA |
| S | REAL | ARRAY | RESULT | 620 | CODE | INTEGER | ARRAY | NP |
| Z | REAL | ARRAY | ARG | 0 | D | REAL | ARRAY | RE |
| L | REAL | ARRAY | DISP | 1130 | DELN | REAL | ARRAY | DI |
| LN1 | REAL | ARRAY | DISP | 7027 | DSIG | REAL | ARRAY | SI |
| | REAL | ARRAY | MATP | 170 | EE | REAL | ARRAY | MA |
| | REAL | ARRAY | SIGM | 3410 | GNM1 | REAL | ARRAY | DI |
| M2 | REAL | ARRAY | DISP | 1 | H | REAL | | BA |
| | INTEGER | | F.P. | 4 | IFLAG | INTEGER | | BA |
| AX | INTEGER | ARRAY | TD | 0 | IMIN | INTEGER | ARRAY | TD |
| | INTEGER | | BASIC2 | 44 | IX | INTEGER | ARRAY | EL |
| | INTEGER | | F.P. | 341 | JMAX | INTEGER | ARRAY | TD |
| IN | INTEGER | ARRAY | TD | 7 | LINC | INTEGER | | BA |
| | INTEGER | | ARG | 1504 | MATRIL | INTEGER | ARRAY | EL |
| XI | INTEGER | | TD | 373 | MAXJ | INTEGER | | TD |
| | INTEGER | | ARG | 375 | NBC | INTEGER | | TD |
| AY | INTEGER | | BASIC2 | 374 | NMTL | INTEGER | | TD |
| NUM | INTEGER | ARRAY | NPDATA | 6 | NTIME | INTEGER | | BA |
| MEL | INTEGER | | BASIC | 3 | NUMLA | INTEGER | | BA |
| MNP | INTEGER | | BASIC | 0 | RO | REAL | ARRAY | MA |
| | REAL | ARRAY | ARG | 140 | SIG | REAL | ARRAY | AR |
| GMA | REAL | ARRAY | SIGM | 30 | TH | REAL | ARRAY | EL |
| ME | REAL | | BASIC2 | 1 | TSUM | REAL | ARRAY | SI |

BROUTINE MODIFY      74/74    OPT=1                                   FTN 4.6+428      7

```
      SUBROUTINE MODIFY
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
C ADD STRESS INCREMENT TO PREVIOUS STRESS
      DO 100 I=1,6
      SIGMA(M,N,I)=SIGMA(M,N,I)+DSIG(I)
  100 CONTINUE
      CALL RESLT
      RETURN
      END
```

MBOLIC REFERENCE MAP (R=1)

NTS
DIFY

| | SN TYPE | | RELOCATION | | | | | |
|---|---|---|---|---|---|---|---|---|
| | REAL | ARRAY | MASS | 14 | ALPHA | REAL | ARRAY | EL |
| | REAL | ARRAY | MASS | 0 | BET | REAL | | BA |
| TA | REAL | ARRAY | ELDATA | 0 | BSUM | REAL | | SI |
| | REAL | ARRAY | RESULT | 2260 | CM | REAL | ARRAY | MA |
| S | REAL | ARRAY | RESULT | 620 | CODE | INTEGER | ARRAY | NP |
| Z | REAL | ARRAY | ARG | 0 | D | REAL | ARRAY | RE |
| L | REAL | ARRAY | DISP | 1130 | DELN | REAL | ARRAY | DI |
| LN1 | REAL | ARRAY | DISP | 7027 | DSIG | REAL | ARRAY | SI |
| | REAL | ARRAY | MATP | 170 | EE | REAL | ARRAY | MA |
| | REAL | ARRAY | SIGM | 3410 | GNM1 | REAL | ARRAY | DI |
| M2 | REAL | ARRAY | DISP | 1 | H | REAL | | BA |
| | INTEGER | | | 4 | IFLAG | INTEGER | | BA |
| AX | INTEGER | ARRAY | TD | 0 | IMIN | INTEGER | ARRAY | TD |
| | INTEGER | | BASIC2 | 44 | IX | INTEGER | ARRAY | EL |
| AX | INTEGER | ARRAY | TD | 310 | JMIN | INTEGER | ARRAY | TD |
| NC | INTEGER | | BASIC2 | 155 | M | INTEGER | | AR |
| TRIL | INTEGER | ARRAY | ELDATA | 372 | MAXI | INTEGER | | TD |
| XJ | INTEGER | | TD | 154 | N | INTEGER | | AR |
| C | INTEGER | | TD | 3 | NLAY | INTEGER | | BA |
| TL | INTEGER | | TD | 1130 | NPNUM | INTEGER | ARRAY | NP |
| IME | INTEGER | | BASIC2 | 2 | NUMEL | INTEGER | | BA |
| MLA | INTEGER | | BASIC | 1 | NUMNP | INTEGER | | BA |
| | REAL | ARRAY | MATP | 24 | S | REAL | ARRAY | AR |
| G | REAL | ARRAY | ARG | 7 | SIGMA | REAL | ARRAY | SI |
| | REAL | ARRAY | ELDATA | 2 | TIME | REAL | | BA |

```
      SUBROUTINE MPLOT
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/R(200),Z(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      REAL X(100),Y(100),TX(2),TY(2),TITLE(20),ZMAX
      READ (5,1000) TITLE,RMAX,ZMAX
      CALL CCP2SY (0.7,0.2,0.2,TITLE,0.0,80)
      CALL CCP1PL (0.7,0.7,-3)
      TX(1)=0.00
      TY(1)=0.00
      TX(2)=RMAX/9.0
      TY(2)=RMAX/9.0
      ZMAX=ZMAX*TY(2)+2.0
      IF (ZMAX.LT.17.0) ZMAX=17.0
      DO 100 J=1,MAXJ
      NSTART=IMIN(J)
      NSTOP=IMAX(J)
      N=0
      DO 101 I=NSTART,NSTOP
      N=N+1
      NP=NPNUM(I,J)
      Y(N)=R(NP)
  101 X(N)=Z(NP)
      CALL CCP6LN (X,Y,N,1,TX,TY)
  100 CONTINUE
      DO 102 I=1,MAXI
      NSTART=JMIN(I)
      NSTOP=JMAX(I)
      N=0
      DO 103 J=NSTART,NSTOP
      N=N+1
      NP=NPNUM(I,J)
      Y(N)=R(NP)
  103 X(N)=Z(NP)
      CALL CCP6LN (X,Y,N,1,TX,TY)
  102 CONTINUE
      CALL CCP1PL (ZMAX,-0.7,-3)
 1000 FORMAT (20A4/2F10.0)
      RETURN
      END
```

MBOLIC REFERENCE MAP (R=1)

```
      FUNCTION NODE(I,J)
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      NODE=0
      DO 100 JJ=1,J
      NSTART=IMIN(JJ)
      NSTOP=IMAX(JJ)
      DO 100 II=NSTART,NSTOP
      NODE=NODE+1
      IF(JJ.EQ.J.AND.II.EQ.I) RETURN
  100 CONTINUE
      RETURN
      END
```

MBOLIC REFERENCE MAP (R=1)

NTS
DE

| SN | TYPE | | RELOCATION | | | | | |
|---|---|---|---|---|---|---|---|---|
| | REAL | ARRAY | MASS | 14 | ALPHA | REAL | ARRAY | EL |
| | REAL | ARRAY | MASS | 0 | BET | REAL | | BA |
| TA | REAL | ARRAY | ELDATA | 0 | BSUM | REAL | | SI |
| | REAL | ARRAY | RESULT | 2260 | CM | REAL | ARRAY | MA |
| S | REAL | ARRAY | RESULT | 620 | CODE | INTEGER | ARRAY | NP |
| Z | REAL | ARRAY | ARG | 0 | D | REAL | ARRAY | RE |
| L | REAL | ARRAY | DISP | 1130 | DELN | REAL | ARRAY | DI |
| LN1 | REAL | ARRAY | DISP | 7027 | DSIG | REAL | ARRAY | SI |
| | REAL | ARRAY | MATP | 170 | EE | REAL | ARRAY | MA |
| | REAL | ARRAY | SIGM | 3410 | GNM1 | REAL | ARRAY | DI |
| M2 | REAL | ARRAY | DISP | 1 | H | REAL | | BA |
| | INTEGER | | F.P. | 4 | IFLAG | INTEGER | | BA |
| | INTEGER | | | 144 | IMAX | INTEGER | ARRAY | TD |
| IN | INTEGER | ARRAY | TD | 5 | IT | INTEGER | | BA |
| | INTEGER | ARRAY | ELDATA | 0 | J | INTEGER | | F |
| | INTEGER | | | 341 | JMAX | INTEGER | ARRAY | TD |
| IN | INTEGER | ARRAY | TD | 7 | LINC | INTEGER | | BA |
| | INTEGER | | ARG | 1504 | MATRIL | INTEGER | ARRAY | EL |
| XI | INTEGER | | TD | 373 | MAXJ | INTEGER | | TD |
| | INTEGER | | ARG | 375 | NBC | INTEGER | | TD |
| AY | INTEGER | | BASIC2 | 374 | NMTL | INTEGER | | TD |
| DE | INTEGER | | | 1130 | NPNUM | INTEGER | ARRAY | NP |

```
      SUBROUTINE POINT(TIME,NUM,TN)
      COMMON/NUMB/NITER,HDAT,TO
      AN=(TIME-TO)/HDAT+1.0
      NUM=AN
      NUM=(AN+NUM+1.5)/2
      TN=TO+(NUM-1)*HDAT
      RETURN
      END
```

BOLIC REFERENCE MAP (R=1)

```
NTS
INT

      SN   TYPE          RELOCATION
           REAL                                  1   HDAT        REAL          NU
TER        INTEGER           NUMB                0   NUM         INTEGER        F
E          REAL               F.P.               0   TN          REAL           F
           REAL              NUMB

OCKS    LENGTH
NB          3

S
 LENGTH                      25B         21
 ED COMMON LENGTH             3B          3
```

```
      SUBROUTINE POINTS
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/PLYLD/SIGY(12),DEPS(6)
      DIMENSION AX(10,20),AY(10,20),BLKANG(12),BLKALF(12)
      EQUIVALENCE (X(1),AX),(Y(1),AY)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     ESTABLISH NODAL POINT INFORMATION
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      NEL=0
      NODSUM=0
      DO 100 J=1,MAXJ
      NSTART=IMIN(J)
      NSTOP=IMAX(J)
      DO 100 I=NSTART,NSTOP
  100 NODSUM=NODSUM+1
      NELSUM=0
      JJMAX=MAXJ-1
      DO 110 JJ=1,JJMAX
      NSTOP=MINO(IMAX(JJ),IMAX(JJ+1))-1
      NSTART=MAXO(IMIN(JJ),IMIN(JJ+1))
      DO 110 II=NSTART,NSTOP
  110 NELSUM=NELSUM+1
      NUMNP=NODSUM
      NUMEL=NELSUM
      DO 120 J=1,MAXJ
      NSTART=IMIN(J)
      NSTOP=IMAX(J)
      DO 120 I=NSTART,NSTOP
      NPNUM(I,J)=NODE(I,J)
      NP=NPNUM(I,J)
      X(NP)=AX(I,J)
  120 Y(NP)=AY(I,J)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     READ AND ASSIGN BOUNDARY CONDITIONS
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     INITIALIZE
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      NUMN=(NUMLA+1)*NUMNP
      DO 130 I=1,NUMN
      CODE(I)=0
  130 CONTINUE
      IF(NBC.EQ.0) GO TO 210
      DO 200 IBCON=1,NBC
      READ(5,1002)N1,N2,ICN
      DO 200 I=N1,N2
```

```
      CODE(I)=ICN
  200 CONTINUE
  210 MPRINT=0
      DO 230 J=1,MAXJ
      NSTART=IMIN(J)
      NSTOP=IMAX(J)
      DO 230 I=NSTART,NSTOP
      NP=NPNUM(I,J)
      IF(MPRINT.NE.0) GO TO 220
      WRITE(6,2000)
      MPRINT=59
  220 MPRINT=MPRINT-1
  230 WRITE(6,2001) I,J,NP,CODE(NP),X(NP),Y(NP)
      DO 310 IMTL=1,NUMLA
      READ (5,1000) MTL,BETA(IMTL),ALPHA(IMTL),SIGY(IMTL)
  310 MATRIL(IMTL)=MTL
      WRITE(6,2003) (CODE(I),I=1,NUMN)
 2003 FORMAT(25I3)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     ESTABLISH ELEMENT INFORMATION
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      JJMAX=MAXJ-1
      N=0
      MTL=1
      KTL=1
      DO 440 JJ=1,JJMAX
      NSTOP=MINO(IMAX(JJ),IMAX(JJ+1))-1
      NSTART=MAXO(IMIN(JJ),IMIN(JJ+1))
      DO 440 II=NSTART,NSTOP
      NEL=NEL+1
  420 I=NPNUM(II,JJ)
      J=I+1
      K=NPNUM(II+1,JJ+1)
      L=K-1
      M=NEL
      IX(M,1)=I
      IX(M,2)=J
      IX(M,3)=K
      IX(M,4)=L
  440 CONTINUE
      IF(NUMNP.GT.2000) WRITE(6,2002)
 1000 FORMAT ( I5,3F10.0)
 1002 FORMAT(2I5,I10)
 2000 FORMAT (59H1       I     J      NP          TYPE      X-ORDINATE     Z-ORDINA
     1TE    )
 2001 FORMAT (2I5,I6,I12,F13.6,F14.6,E26.7,E24.7,E24.7)
 2002 FORMAT (35H  BAD INPUT - TOO MANY NODAL POINTS)
      RETURN
      END
```

MBOLIC REFERENCE MAP (R=1)

```
      SUBROUTINE QUAD
      INTEGER CODE
      REAL NUSN,NUTN,NUTS,NUNS,NUNT,NUST
      DIMENSION DUMMY(6,6),DUMMY1(6,6)
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
       IF(NTIME.EQ.1) GO TO 10
      READ(1) ((CRZ(I,J),J=1,6),I=1,6)
       GO TO 151
   10  CONTINUE
      I1=IX(N,1)
      J1=IX(N,2)
      K1=IX(N,3)
      L1=IX(N,4)
      MTYPE=MATRIL(NLAY)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     INTERPOLATE MATERIAL PROPERTIES
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DO 100 I=1,9
  100 EE(I)=E(I,MTYPE)
      DO 110 I=1,6
      DO 110 J=1,6
      CNS(I,J)=0.00
      C(I,J)=0.00
  110 D(I,J)=0.00
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     FORM STRESS-STRAIN RELATIONSHIP IN N-S-T SYSTEM
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      NUNS=EE(4)
      NUNT=EE(5)
      NUST=EE(6)
      NUSN=(EE(2)*NUNS)/EE(1)
      NUTN=(EE(3)*NUNT)/EE(1)
      NUTS=(EE(3)*NUST)/EE(2)
      DIV=1.00-NUNS*NUSN-NUST*NUTS-NUNT*NUTN-NUSN*NUNT*NUTS
     1-NUNS*NUTN*NUST
      CNS(1,1)=EE(1)*(1.00-NUST*NUTS)/DIV
      CNS(1,2)=EE(2)*(NUNS+NUNT*NUTS)/DIV
      CNS(1,3)=EE(3)*(NUNT+NUNS*NUST)/DIV
      CNS(2,1)=CNS(1,2)
      CNS(2,2)=EE(2)*(1.00-NUNT*NUTN)/DIV
      CNS(2,3)=EE(3)*(NUST+NUSN*NUNT)/DIV
      CNS(3,1)=CNS(1,3)
      CNS(3,2)=CNS(2,3)
      CNS(3,3)=EE(3)*(1.00-NUNS*NUSN)/DIV
      CNS(4,4)=EE(7)
      CNS(5,5)=EE(8)
```

```
      CNS(6,6)=EE(9)
C     SET UP STRAIN TRANSFORM TO N-S-T SYSTEM
      SINA=SIN(ALPHA(M))
      COSA=COS(ALPHA(M))
      S2=SINA**2
      C2=COSA**2
      SC=SINA*COSA
      D(1,1)=C2
      D(1,3)=S2
      D(1,6)=-SC
      D(2,1)=S2
      D(2,3)=C2
      D(2,6)=SC
      D(3,2)=1.00
      D(4,1)=2.00*SC
      D(4,3)=-2.00*SC
      D(4,6)=C2-S2
      D(5,4)=SINA
      D(5,5)=COSA
      D(6,4)=COSA
      D(6,5)=-SINA
C     SET UP STRAIN TRANSFORMATION TO R-Z-T SYSTEM
      SINB=SIN(BETA(M))
      COSB=COS(BETA(M))
      S2=SINB**2
      C2=COSB**2
      SC=SINB*COSB
      C(1,1)=S2
      C(1,2)=C2
      C(1,4)=SC
      C(2,1)=C2
      C(2,2)=S2
      C(2,4)=-SC
      C(3,3)=1.00
      C(4,1)=-2.00*SC
      C(4,2)=2.00*SC
      C(4,4)=S2-C2
      C(5,5)=SINB
      C(5,6)=-COSB
      C(6,5)=COSB
      C(6,6)=SINB
C     CALCULATE CRZ MATRIX
      DO 120 I=1,6
      DO 120 J=1,6
      DUMMY(I,J)=0.00
      DO 120 K=1,6
  120 DUMMY(I,J)=DUMMY(I,J)+D(I,K)*C(K,J)
      DO 130 I=1,6
      DO 130 J=1,6
      DUMMY1(I,J)=0.00
      DO 130 K=1,6
  130 DUMMY1(I,J)=DUMMY1(I,J)+CNS(I,K)*DUMMY(K,J)
      DO 140 I=1,6
      DO 140 J=1,6
      DUMMY(I,J)=0.00
      DO 140 K=1,6
  140 DUMMY(I,J)=DUMMY(I,J)+D(K,I)*DUMMY1(K,J)
```

```
          DO 150 I=1,6
          DO 150 J=1,6
          CRZ(I,J)=0.00
          DO 150 K=1,6
      150 CRZ(I,J)=CRZ(I,J)+C(K,I)*DUMMY(K,J)
          WRITE(1) ((CRZ(I,J),J=1,6),I=1,6)
      151  CONTINUE
C
          DO 200 I=1,4
          MM= IX(N,I)
          XXX(I)= X(MM)
          XXX(I+4)= X(MM)
          YYY(I)= Y(MM)
      200 YYY(I+4)= Y(MM)
          XXX(9)= (XXX(1)+ XXX(2)+ XXX(3)+ XXX(4))/4.00
          YYY(9)= (YYY(1)+ YYY(2)+ YYY(3)+ YYY(4))/4.00
          XXX(10)= XXX(9)
          YYY(10)= YYY(9)
          DO 250 I=1,8
          CM(I)=0.00
      250 CONTINUE
          DO 900 IA=1,24
      900 S(IA)= 0.00
          CALL TRISTF(1,2)
          CALL TRISTF(2,3)
          CALL TRISTF(3,4)
          CALL TRISTF(4,1)
          RETURN
          END
```

MBOLIC REFERENCE MAP (R=1)

NTS
AD

| SN | TYPE | | RELOCATION | | | | | |
|---|---|---|---|---|---|---|---|---|
| | REAL | ARRAY | MASS | 14 | ALPHA | REAL | ARRAY | EL |
| | REAL | ARRAY | MASS | 0 | BET | REAL | | BA |
| TA | REAL | ARRAY | ELDATA | 0 | BSUM | REAL | | SI |
| | REAL | ARRAY | RESULT | 2260 | CM | REAL | ARRAY | MA |
| S | REAL | ARRAY | RESULT | 620 | CODE | INTEGER | ARRAY | NP |
| SA | REAL | | | 515 | COSB | REAL | | |
| Z | REAL | ARRAY | ARG | 512 | C2 | REAL | | |
| | REAL | ARRAY | RESULT | 2260 | DEL | REAL | ARRAY | DI |
| LN | REAL | ARRAY | DISP | 0 | DELN1 | REAL | ARRAY | DI |
| V | REAL | | | 7027 | DSIG | REAL | ARRAY | SI |
| MMY | REAL | ARRAY | | 565 | DUMMY1 | REAL | ARRAY | |
| | REAL | ARRAY | MATP | 170 | EE | REAL | ARRAY | MA |
| | REAL | ARRAY | SIGM | 3410 | GNM1 | REAL | ARRAY | DI |
| M2 | REAL | ARRAY | DISP | 1 | H | REAL | | BA |
| | INTEGER | | | 520 | IA | INTEGER | | |
| LAG | INTEGER | | BASIC2 | 144 | IMAX | INTEGER | ARRAY | TD |
| IN | INTEGER | ARRAY | TD | 5 | IT | INTEGER | | BA |

```
      SUBROUTINE RESLT
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      RETURN
      END
```

MBOLIC REFERENCE MAP (R=1)

```
NTS
SLT

     SN  TYPE              RELOCATION
         REAL         ARRAY     MASS            14  ALPHA     REAL      ARRAY    EL
         REAL         ARRAY     MASS             0  BET       REAL               BA
TA       REAL         ARRAY     ELDATA           0  BSUM      REAL               SI
         REAL         ARRAY     RESULT        2260  CM        REAL      ARRAY    MA
S        REAL         ARRAY     RESULT         620  CODE      INTEGER   ARRAY    NP
Z        REAL         ARRAY     ARG              0  D         REAL      ARRAY    RE
L        REAL         ARRAY     DISP          1130  DELN      REAL      ARRAY    DI
LN1      REAL         ARRAY     DISP          7027  DSIG      REAL      ARRAY    SI
         REAL         ARRAY     MATP           170  EE        REAL      ARRAY    MA
         REAL         ARRAY     SIGM          3410  GNM1      REAL      ARRAY    DI
M2       REAL         ARRAY     DISP             1  H         REAL               BA
LAG      INTEGER                BASIC2         144  IMAX      INTEGER   ARRAY    TD
IN       INTEGER      ARRAY     TD               5  IT        INTEGER            BA
         INTEGER      ARRAY     ELDATA         341  JMAX      INTEGER   ARRAY    TD
IN       INTEGER      ARRAY     TD               7  LINC      INTEGER            BA
         INTEGER                ARG           1504  MATRIL    INTEGER   ARRAY    EL
XI       INTEGER                TD             373  MAXJ      INTEGER            TD
         INTEGER                ARG            375  NBC       INTEGER            TD
AY       INTEGER                BASIC2         374  NMTL      INTEGER            TD
NUM      INTEGER      ARRAY     NPDATA           6  NTIME     INTEGER            BA
MEL      INTEGER                BASIC            3  NUMLA     INTEGER            BA
MNP      INTEGER                BASIC            0  RO        REAL      ARRAY    MA
         REAL         ARRAY     ARG            140  SIG       REAL      ARRAY    AR
GMA      REAL         ARRAY     SIGM            30  TH        REAL      ARRAY    EL
ME       REAL                   BASIC2           1  TSUM      REAL      ARRAY    SI
L        REAL                   BASIC            0  X         REAL      ARRAY    NP
         REAL         ARRAY     ARG             54  XX        REAL      ARRAY    AR
X        REAL         ARRAY     ARG            310  Y         REAL      ARRAY    NP
         REAL         ARRAY     ARG             12  YYY       REAL      ARRAY    AR
```

```
      SUBROUTINE STIFF
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATR/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/MASS1/XMINV(600)
      IFLAG=1
      IF(NTIME.EQ.1) GO TO 75
      DO 50 I=1,IT
      GNM2(I)=GNM1(I)
      GNM1(I)=B(I)
   50 CONTINUE
  75  CONTINUE
      DO 100 I=1,IT
      B(I)= 0.00
  100 CONTINUE
      IF(NTIME.GT.1)GO TO 102
      DO 101 I=1,IT
      A(I)= 0.00
  101 CONTINUE
  102 CONTINUE
      REWIND 1
      REWIND 2
      REWIND 3
      DO 340 M=1,NUMLA
      NLAY=M
      DO 340 N=1,NUMEL
      CALL QUAD
      DO 340 I=1,4
      II = 3*IX(N,I)+ 3*(M-1)*NUMNP
      J= II-2
      DO 340 K=J,II
      JJ=K-II+3*I
      B(K)= B(K)+S(JJ)
      KK= K+3*NUMNP
      B(KK)= B(KK)+S(JJ+12)
      IF(NTIME.GT.1)GO TO 340
      A(K)=A(K)+CM(I)
      A(KK)=A(KK)+CM(I+4)
  340 CONTINUE
      CALL LOAD
      DO 400I=1,IT
  400 B(I)=-B(I)+F(I)
      RETURN
      END
```

```
      SUBROUTINE STRESS
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/NPR/NPRINT
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CR2(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/DELT/XDEL(4,18)
      COMMON/DELTRI/DELTA(4,18)
      COMMON/PLYLD/SIGY(12),DEPS(6)
      DIMENSION DDEL(600)
      IF(NTIME.EQ.1) GO TO 98
      IF((NTIME/NPRINT)*NPRINT.NE.NTIME) GO TO 99
   98 CONTINUE
      WRITE(6,1003)NTIME
 1003 FORMAT(" DEL FROM SUBROUTINE STRESS,CYCLE=",I5)
      WRITE(6,1001)(DEL(I),I=1,IT)
 1001 FORMAT(9E12.6)
   99 CONTINUE
C CALCULATE CHANGES IN DISPLACEMENTS
      DO 100 I=1,IT
      DDEL(I)=DEL(I)-DELN(I)
  100 CONTINUE
      DO 101 I=1,4
      DO 101 J=1,18
      XDEL(I,J)=0.00
      DELTA(I,J)=0.00
  101 CONTINUE
      IFLAG=2
      REWIND 1
      REWIND 2
      REWIND 3
      DO 200 M=1,NUMLA
      NLAY=M
      DO 200 N=1,NUMEL
      DO 150 I=1,4
      I1=I+1
      IF(I.EQ.4) I1=1
      MM=IX(N,I)
      MM1=IX(N,I1)
      II=3*MM+3*(M-1)*NUMNP
      II1=3*MM1+3*(M-1)*NUMNP
      J=II-2
      J1=II1-2
      DO 140 K=J,II
      IK=K-J+1
      XDEL(I,IK)=DDEL(K)
      DELTA(I,IK)=DELN(K)
      KK=K+3*NUMNP
```

```
      XDEL(I,IK+6)=DDEL(KK)
      DELTA(I,IK+6)=DELN(KK)
  140 CONTINUE
      DO 145 K=J1,II1
      IK=K-J1+1
      XDEL(I,IK+3)=DDEL(K)
      DELTA(I,IK+3)=DELN(K)
      KK=K+3*NUMNP
      XDEL(I,IK+9)=DDEL(KK)
      DELTA(I,IK+9)=DELN(KK)
  145 CONTINUE
  150 CONTINUE
C FIND DISPLACEMENT, BY AVERAGING, AT CENTER OF EACH ELEMENT
      XDEL(1,13)=(XDEL(1,1)+XDEL(2,1)+XDEL(3,1)+XDEL(4,1))/4.00
      XDEL(1,14)=(XDEL(1,2)+XDEL(2,2)+XDEL(3,2)+XDEL(4,2))/4.00
      XDEL(1,15)=(XDEL(1,3)+XDEL(2,3)+XDEL(3,3)+XDEL(4,3))/4.00
      XDEL(1,16)=(XDEL(1,7)+XDEL(2,7)+XDEL(3,7)+XDEL(4,7))/4.00
      XDEL(1,17)=(XDEL(1,8)+XDEL(2,8)+XDEL(3,8)+XDEL(4,8))/4.00
      XDEL(1,18)=(XDEL(1,9)+XDEL(2,9)+XDEL(3,9)+XDEL(4,9))/4.00
      DELTA(1,13)=(DELTA(1,1)+DELTA(2,1)+DELTA(3,1)+DELTA(4,1))
     1/4.00
      DELTA(1,14)=(DELTA(1,2)+DELTA(2,2)+DELTA(3,2)+DELTA(4,2))
     1/4.00
      DELTA(1,15)=(DELTA(1,3)+DELTA(2,3)+DELTA(3,3)+DELTA(4,3))
     1/4.00
      DELTA(1,16)=(DELTA(1,7)+DELTA(2,7)+DELTA(3,7)+DELTA(4,7))
     1/4.00
      DELTA(1,17)=(DELTA(1,8)+DELTA(2,8)+DELTA(3,8)+DELTA(4,8))
     1/4.00
      DELTA(1,18)=(DELTA(1,9)+DELTA(2,9)+DELTA(3,9)+DELTA(4,9))
     1/4.00
C CALCULATE AVERAGE STRAIN
      DO 175 I=2,4
      DO 175 J=13,18
      XDEL(I,J)=XDEL(1,J)
      DELTA(I,J)=DELTA(1,J)
  175 CONTINUE
  179 BSUM=0.00
      DO 180 I=1,6
      TSUM(I)=0.00
  180 CONTINUE
      CALL QUAD
      DO 190 I=1,6
  190 DEPS(I)=TSUM(I)/BSUM
C CALCULATE STRESS INCREMENT
      DO 195 I=1,6
      DSIG(I)=0.00
      DO 195 J=1,6
  195 DSIG(I)= DSIG(I)+CRZ(I,J)*DEPS(J)
        CALL YIELD
      CALL MODIFY
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRISTF(II,JJ)
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LINC
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/DELT/XDEL(4,18)
      COMMON/DELTRI/DELTA(4,18)
      DIMENSION Q(6,18),C(18,18),XY(9,9),SAVE(18,3),BT(18,6),BTSIG(18),
     1CBAR(10,18),Z(10,10),             CBD(10),CC(10,4),BLT(18,4),
     2BADD(18,6)
      DIMENSION BLTSIG(18),BSIG(18)
      DIMENSION DUM(10,10)
      NLA=M
      NEL=N
      DO 50 I=1,6
   50 SIG(I)=SIGMA(M,N,I)
      THICK=TH(NLAY)
      ISUB= 9
      XX(1)= XXX(II)
      XX(2)= XXX(JJ)
      XX(3)= XXX(ISUB)
      YY(1)= YYY(II)
      YY(2)= YYY(JJ)
      YY(3)= YYY(ISUB)
       IF(NTIME.EQ.1) GO TO 40
      READ(2) ((BT(I,J),J=1,6),I=1,18)
       GO TO 801
   40  CONTINUE
  100 CALL INTER
  150 CONTINUE
      DO 300 IA=1,6
      DO 300 JA=1,18
  300 Q(IA,JA)=0.00
      Q(1,2)= XI(1)* THICK
      Q(1,11)= XI(1)*((THICK)**2)/2.00
      Q(3,13)= XI(3)* THICK
      Q(5,14)= XI(2)* THICK
      Q(2,6)= Q(1,2)
      Q(3,16)= Q(1,2)
      Q(3,17)= Q(5,14)
      Q(4,3)= Q(1,2)
      Q(4,5)= Q(1,2)
      Q(5,9)= Q(1,2)
      Q(5,13)= Q(1,2)
      Q(6,8)= Q(1,2)
      Q(5,10)= Q(1,2)
      Q(2,15)= Q(1,11)
      Q(4,12)= Q(1,11)
      Q(4,14)= Q(1,11)
```

```
      Q(5,18)= Q(1,11)
      Q(6,17)= Q(1,11)
      Q(5,15)= Q(3,18)
      Q(5,12)= Q(3,18)
      Q(6,11)= Q(5,14)
C
C THE FOLLOWING EXPRESSION IS ACTUALLY TWICE THE AREA OF THE TRIANGLE.
C
      AREA= XX(1)*YY(2)-XX(2)*YY(1)+XX(2)*YY(3)-XX(3)*YY(2)+XX(3)*YY(1)-
     1XX(1)*YY(3)
C
C     ZERO C MATRIX
C
      DO 400 I=1,18
      DO 400 J=1,18
  400 C(I,J)= 0.00
      FAC=1/AREA
      M=1
      N=1
      I= II
      J= JJ
      GO TO 250
   75 CONTINUE
      FAC=1/(AREA*THICK)
      M=10
      N=10
      I=II+4
      J=JJ+4
C
C   FILL C INVERSE BY PARTS
C
  250 C(M,N)= (XXX(J)* YYY(ISUB)- XXX(ISUB)* YYY(J))*FAC
      C(M,N+3)= (XXX(ISUB)* YYY(I)- XXX(I)* YYY(ISUB))*FAC
      C(M,N+6)= (XXX(I)* YYY(J)- XXX(J)* YYY(I))*FAC
      C(M+1,N)=(YYY(J)- YYY(ISUB))*FAC
      C(M+1,N+3)=(YYY(ISUB)- YYY(I))*FAC
      C(M+1,N+6)=(YYY(I)- YYY(J))*FAC
      C(M+2,N)=(XXX(ISUB)- XXX(J))*FAC
      C(M+2,N+3)=(XXX(I)- XXX(ISUB))*FAC
      C(M+2,N+6)=(XXX(J)- XXX(I))*FAC
      C(M+3,N+1)= C(M,N)
      C(M+3,N+4)= C(M,N+3)
      C(M+3,N+7)= C(M,N+6)
      C(M+4,N+1)= C(M+1,N)
      C(M+4,N+4)= C(M+1,N+3)
      C(M+4,N+7)= C(M+1,N+6)
      C(M+5,N+1)= C(M+2,N)
      C(M+5,N+4)= C(M+2,N+3)
      C(M+5,N+7)= C(M+2,N+6)
      C(M+6,N+2)= C(M,N)
      C(M+6,N+5)= C(M,N+3)
      C(M+6,N+8)= C(M,N+6)
      C(M+7,N+2)= C(M+1,N)
      C(M+7,N+5)= C(M+1,N+3)
      C(M+7,N+8)= C(M+1,N+6)
      C(M+8,N+2)= C(M+2,N)
      C(M+8,N+5)= C(M+2,N+3)
```

```
      C(M+8,N+8)= C(M+2,N+6)
      IF(ISUB.EQ.10) GO TO 200
      ISUB=-10
      GO TO 75
  200 M=NLA
      N=NEL
C
C     FILL XY MATRIX
C
      DO 600 IJ=1,9
      DO 600 JI=1,9
  600 XY(IJ,JI)= 0.00
      XY(1,1)=  1.00
      XY(1,2)=  XXX(I)
      XY(1,3)=  YYY(I)
      XY(4,2)=  XXX(J)
      XY(4,3)=  YYY(J)
      XY(7,2)=  XXX(10)
      XY(7,3)=  YYY(10)
      XY(2,4)=  XY(1,1)
      XY(3,7)=  XY(1,1)
      XY(4,1)=  XY(1,1)
      XY(5,4)=  XY(1,1)
      XY(6,7)=  XY(1,1)
      XY(7,1)=  XY(1,1)
      XY(8,4)=  XY(1,1)
      XY(9,7)=  XY(1,1)
      XY(2,5)=  XY(1,2)
      XY(3,8)=  XY(1,2)
      XY(2,6)=  XY(1,3)
      XY(3,9)=  XY(1,3)
      XY(5,5)=  XY(4,2)
      XY(6,8)=  XY(4,2)
      XY(5,6)=  XY(4,3)
      XY(6,9)=  XY(4,3)
      XY(8,5)=  XY(7,2)
      XY(9,8)=  XY(7,2)
      XY(8,6)=  XY(7,3)
      XY(9,9)=  XY(7,3)
C
C     COMPUTING THE LOWER LEFT QUADRANT OF THE C MATRIX
C
      DO 620 I=1,9
      DO 620 J=1,9
      DUM(I,J)=0.00
      DO 620 K=1,9
      DUM(I,J)=DUM(I,J)+XY(I,K)*C(K,J)
  620 CONTINUE
      DO 630 I=10,18
      DO 630 J=1,9
      C(I,J)=0.00
      DO 630 K=1,9
      K9=K+9
      C(I,J)=C(I,J)-C(I,K9)*DUM(K,J)
  630 CONTINUE
C
C   REARRANGE THE DISPLACEMENT MATRIX
```

```
      C
            DO 730 IA=1,18
            DO 730 JA=1,3
            JB= JA+ 6
        730 SAVE(IA,JA)= C(IA,JB)
            DO 740 IA=1,18
            DO 740 JA=7,12
            JB=JA+3
        740 C(IA,JA)= C(IA,JB)
            DO 750 IA=1,18
            DO 750 JA=1,3
            JB=JA+12
        750 C(IA,JB)= SAVE(IA,JA)
      C
      C   CALCULATE THE BO TRANSPOSED MATRIX
      C
            DO 800 JA=1,18
            DO 800 IA=1,6
            BT(JA,IA)=0.00
            DO 800 MA=1,18
        800 BT(JA,IA)=BT(JA,IA)+C(MA,JA)*Q(IA,MA)
             WRITE(2)((BT(I,J),J=1,6),I=1,18)
        801  CONTINUE
      C
      C   CALCULATE THE BOT * STRESS
      C
            DO 810 IA=1,18
            BTSIG(IA)=0.00
            DO 810 NA=1,6
        810 BTSIG(IA)=BTSIG(IA)+BT(IA,NA)*SIG(NA)
      C
      C  ADD THE NONLINEAR TERMS
      C
      C
      C   CALCULATE THE CBAR MATRIX
      C
             IF(NTIME.EQ.1) GO TO 815
             IF((NTIME/20 )*20.EQ. NTIME) GO TO 815
              READ(3) ((BLT(I,J),J=1,4),I=1,18)
             GO TO 661
        815 CONTINUE
            DO 820 IA=1,10
            DO 820 JA=1,18
            IB= IA+7
            IF(IA.EQ.9)IB=17
            IF(IA.EQ.10)IB=18
        820 CBAR(IA,JA)= C(IB,JA)
      C
      C   CALCULATE THE ZBAR MATRIX
      C
            DO 830 IA=1,10
            DO 830 JA=1,10
        830 Z(IA,JA)= 0.00
            Z(1,1)= XI(1)*THICK
            Z(2,2)= Z(1,1)
            Z(3,3)= Z(1,1)
            Z(6,6)= Z(1,1)
```

```
      Z(3,4)= XI(2)*THICK
      Z(4,3)= Z(3,4)
      Z(6,7)= Z(3,4)
      Z(7,6)= Z(3,4)
      Z(9,9)= (THICK**3)/3*XI(1)
      Z(10,10)= Z(9,9)
      Z(1,9)= (THICK**2)/2*XI(1)
      Z(2,10)= Z(1,9)
      Z(9,1)= Z(1,9)
      Z(10,2)= Z(1,9)
      Z(3,5)= XI(3)* THICK
      Z(5,3)= Z(3,5)
      Z(6,8)= Z(3,5)
      Z(8,6)= Z(3,5)
      Z(4,4)= XI(5)* THICK
      Z(7,7)= Z(4,4)
      Z(5,5)= XI(6)* THICK
      Z(8,8)= Z(5,5)
      Z(4,5)= XI(4)* THICK
      Z(5,4)= Z(4,5)
      Z(7,8)= Z(4,5)
      Z(8,7)= Z(4,5)
C
C     CALCULATING THE CC MATRIX
C
      DO 840 IA=1,10
      CBD(IA)= 0.00
      DO 840 JA=1,18
  840 CBD(IA)= CBD(IA)+ CBAR(IA,JA)* DELTA(II,JA)
      DO 850 IA=1,10
      DO 850 JA=1,4
  850 CC(IA,JA)=0.00
      CC(1,1)= CBD(1)
      CC(2,4)= CBD(1)
      CC(1,4)= CBD(2)
      CC(2,2)= CBD(2)
      CC(3,3)= CBD(3)
      CC(4,3)= CBD(4)
      CC(5,3)= CBD(5)
      CC(6,3)= CBD(6)
      CC(7,3)= CBD(7)
      CC(8,3)= CBD(8)
      CC(9,1)= CBD(9)
      CC(10,4)= CBD(9)
      CC(9,4)= CBD(10)
      CC(10,2)= CBD(10)
      DO 855 I=1,10
      DO 855 J=1,4
      DUM(I,J)=0.00
      DO 855 K=1,10
      DUM(I,J)=DUM(I,J)+Z(I,K)*CC(K,J)
  855 CONTINUE
      DO 860 I=1,18
      DO 860 J=1,4
      BLT(I,J)=0.00
      DO 860 K=1,10
      BLT(I,J)=BLT(I,J)+CBAR(K,I)*DUM(K,J)
```

```
 860  CONTINUE
      WRITE(3) ((BLT(I,J),J=1,4),I=1,18)
  861  CONTINUE
      DO 870 IA=1,18
      BLTSIG(IA)= 0.00
      DO 870 JA=1,4
 870  BLTSIG(IA)= BLTSIG(IA)+ BLT(IA,JA)* SIG(JA)
      DO 880 IA=1,18
      BSIG(IA)= BSIG(IA)+ BLTSIG(IA)
 880  CONTINUE
C  ADDING THE TRIANGULAR BSIG MATRICES TO FORM THE QUADRILATERAL MATRIX
      IF(IFLAG.NE.2) GO TO 1600
C EVALUATE STRAIN INCREMENT
      BSUM=BSUM+AREA/2.0*THICK
      DO 525 I=1,18
      DO 525 J=1,4
      BADD(I,J)= BT(I,J)+ BLT(I,J)
 525  CONTINUE
      DO 526 I=1,18
      DO 526 J=5,6
 526  BADD(I,J)= BT(I,J)
      DO 550 I=1,6
      DO 550 J=1,18
 550  TSUM(I)=TSUM(I)+BADD(J,I)*XDEL(II,J)
      GO TO 1000
1600  CONTINUE
      K= 3*II-2
      L= 3*JJ-2
      DO 910 MA=1,2
      MJ= (MA-1)* 6
      NJ= (MA-1)* 12
      DO 920 IA=1,3
      JA= IA-1
      S(K+ NJ+ JA)= S(K+ NJ+ JA)+ BSIG(MJ+ IA)
 920  S(L+ NJ+ JA)= S(L+ NJ+ JA)+ BSIG(MJ+ IA+ 3)
      DO 910 IA=1,3
      DO 910 JA=1,4
      IJ= (JA-1)*3
 910  S(NJ+IJ+IA)= S(NJ+IJ+IA)+ BSIG(12+(MA-1)*3+IA)/4.0
      IF(NTIME.GT.1) GO TO 1000
C ASSEMBLE ELEMENT MASS MATRIX
      DEN=RO(MATRIL(NLAY))*THICK
      XMASS=AREA*DEN/8.00
      CM(II)=CM(II)+XMASS
      CM(JJ)=CM(JJ)+XMASS
      CM(II+4)=CM(II)
      CM(JJ+4)=CM(JJ)
1000  CONTINUE
       RETURN
      END
```

IBOLIC REFERENCE MAP (R=1)

```
      SUBROUTINE YIELD
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA
      COMMON/MATP/RO(12),E(9,12),EE(9)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,TIME,NLAY,IFLAG,IT,NTIME,LING
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/PLYLD/SIGY(12),DEPS(6)
      DIMENSION SIGT(6),SIGND(6),SIGN1D(6),SIGD(6)
      DO 15 I=1,6
      SIGT(I)=SIGMA(M,N,I)+DSIG(I)
   15 CONTINUE
C    CALCULATE THE DEVIATORIC COMPONENTS OF STRESS
      SKNN1=SIGT(1)+SIGT(2)+SIGT(3)
      DO 30 I=1,3
      SIGD(I)=SIGT(I)-SKNN1/3.0
   30 CONTINUE
      DO 31 I=4,6
      SIGD(I)=SIGT(I)
   31 CONTINUE
C       TEST FOR YIELDING
        YLD = SIGD(1)**2+SIGD(2)**2 +SIGD(3)**2 + 2.*(SIGD(4)**2 +
     1       SIGD(5)**2 +SIGD(6)**2) -(2./3.) *SIGY(M)**2
      IF(YLD.LT.0.00) GO TO 100
C     PLASTIC FLOW HAS OCCURRED IN LAYER=M,ELEMENT=N
C     CORRECT STRESSES FOR PLASTICITY
C     CALCULATE LAMBDA-BAR
C     CALCULATE A,B,C----AQ,BQ,CQ
      AQ=SIGD(1)**2+SIGD(2)**2+SIGD(3)**2 +2.*(SIGD(4)**2
     1    +SIGD(5)**2+SIGD(6)**2)
      BQ= SIGT(1)*SIGD(1)+SIGT(2)*SIGD(2)+SIGT(3)*SIGD(3)+2.*(
     1    SIGT(4)*SIGD(4)+SIGT(5)*SIGD(5)+SIGT(6)*SIGD(6))
        CQ=YLD
      DISC = BQ**2 -AQ*CQ
      IF(DISC.LT. 0.0) DISC=0.00
  400 XLMSTR= CQ/ (BQ + SQRT(DISC))
C     CALCULATE PLASTIC STRESS INCREMENT
      DO 40 I=1,6
      DSIG(I)=DSIG(I)-SIGD(I)*XLMSTR
   40 CONTINUE
  300 CONTINUE
  100 CONTINUE
      RETURN
      END
```

DISTRIBUTION LIST

| No. of Copies | Organization |
|---|---|
| 12 | Commander<br>Defense Documentation Center<br>ATTN: DDC-TCA<br>Cameron Station<br>Alexandria, VA 22314 |
| 1 | Director of Defense Research and Engineering<br>ATTN: Tech Lib, Rm 3E-1039<br>Washington, DC 20301 |
| 1 | Commander<br>US Army Materiel Command<br>ATTN: DRCDMA-ST<br>5001 Eisenhower Avenue<br>Alexandria, VA 22333 |
| 1 | Commander<br>US Army Aviation Systems Command<br>ATTN: DRSAV-E<br>12th and Spruce Streets<br>St. Louis, MO 63166 |
| 1 | Director<br>US Army Air Mobility Research and Development Laboratory<br>Ames Research Center<br>Moffett Field, CA 94035 |
| 1 | Commander<br>US Army Electronics Command<br>ATTN: DRSEL-RD<br>Fort Monmouth, NJ 07703 |
| 3 | Commander<br>US Army Missile R&D Command<br>ATTN: DRDMI-R<br>DRDMI-RFL, Mr. B. Cobb<br>DRDMI-RL, Mr. N. Comus<br>Redstone Arsenal, AL 35809 |
| 1 | Commander<br>US Army Tank Automotive Development Command<br>ATTN: DRDTA-RWL<br>Warren, MI 48090 |
| 2 | Commander<br>US Army Mobility Equipment Research & Development Command<br>ATTN: Tech Docu Cen, Bldg. 315<br>DRSME-RZT<br>Fort Belvoir, VA 22060 |
| 2 | Commander<br>US Army Armament Materiel Readiness Command<br>ATTN: DRSAR-RDT, J. Salamon<br>Dr. L. Johnson<br>Rock Island, IL 61202 |
| 6 | Commander<br>USA Armament Research & Development Command<br>ATTN: SARPA-FR-E, Dr. N. Clark<br>Mr. G. Randers-Pherson<br>SARPA-AD-S, S. Polanski<br>SARPA-FR-M-MA, M. Eig<br>SARPA-FR-S-R, C. Larsen<br>Tech Lib<br>Dover, New Jersey 07801 |
| 2 | Commander<br>US Army Frankford Arsenal<br>ATTN: SARFA-C2500<br>SARFA-L3200<br>Mr. P. Gordon<br>Philadelphia, PA 19137 |
| 3 | Commander<br>US Army Watervliet Arsenal<br>ATTN: SARWV-RPD-SE<br>Mr. M. Dale<br>Dr. J. Santini<br>SARWV-PSD, Dr. G.D'Andrea<br>Watervliet, NY 12189 |

# DISTRIBUTION LIST

| No. of Copies | Organization |
|---|---|
| 1 | Commander<br>US Army Harry Diamond Labs<br>ATTN: DRXDO-TI<br>2800 Powder Mill Road<br>Adelphi, MD 20783 |
| 4 | Commander<br>US Army Materials and<br>Mechanics Research Center<br>ATTN: DRXMR-ATL, W. Woods<br>DRXMR-T, J. Mescall<br>DRXMR-TM, E. Leone<br>Tech Lib<br>Watertown, MA 02172 |
| 1 | Director<br>US Army TRADOC Systems<br>Analysis Activity<br>ATTN: ATAA-SA<br>White Sands Missile Range<br>New Mexico 88002 |
| 1 | Assistant Secretary of the<br>Army (R&D)<br>ATTN: Asst for Research<br>Washington, DC 20310 |
| 2 | HQDA (DAMA-ZA; DAMA-AR)<br>Washington, DC 20310 |
| 1 | HQDA (DAMA-CSM, LTC N. O.<br>Connor, Jr.)<br>Washington, DC 20310 |
| 1 | Commander<br>US Army Research Office<br>P.O. Box 12211<br>Research Triangle Park<br>North Carolina 27709 |
| 2 | Commander<br>US Naval Air Systems Command<br>ATTN: Code AIR-310<br>Code AIR-350<br>Washington, DC 20360 |
| 1 | Commander<br>US Naval Ordnance Systems Command<br>ATTN: Code ORD-0332<br>Washington, DC 20360 |
| 2 | Chief of Naval Research<br>ATTN: Code 427<br>Code 470<br>Department of the Navy<br>Washington, DC 20325 |
| 3 | Commander<br>US Naval Surface Weapons Center<br>ATTN: Dr. H. Sternburg<br>Dr. Walker<br>Code 730, Lib<br>Silver Spring, MD 20910 |
| 2 | Commander<br>US Naval Surface Weapons Center<br>ATTN: Code GWD, K. Bannister<br>Tech Lib<br>Dahlgren, VA 22448 |
| 1 | Commander<br>US Naval Weapons Center<br>ATTN: Code 45, Tech Lib<br>China Lake, CA 93555 |
| 1 | Commander<br>US Naval Research Laboratory<br>Washington, DC 20375 |
| 1 | USAF (AFRDDA)<br>Washington, DC 20330 |
| 1 | AFSC (SDW)<br>Andrews AFB<br>Washington, DC 20331 |
| 1 | US Air Force Academy<br>ATTN: Code FJS-RL(NC)<br>Tech Lib<br>Colorado Springs, CO 80840 |

DISTRIBUTION LIST

| No. of Copies | Organization |
|---|---|
| 1 | Commander<br>Hill Air Force Base<br>ATTN: Code AMA<br>Code MMECB<br>Utah 84401 |
| 1 | AFWL (SUL, LT Tennant)<br>Kirtland AFB, NM 87116 |
| 1 | AFLC (MMWMC)<br>Wright-Patterson AFB, OH 45433 |
| 1 | AFAL (AVW)<br>Wright-Patterson AFB, OH 45433 |
| 1 | Director<br>US Bureau of Mines<br>ATTN: Mr. R. Watson<br>4800 Forbes Street<br>Pittsburgh, PA 15213 |
| 1 | Director<br>National Aeronautics and Space Administration<br>Langley Research Center<br>Langley Station<br>Hampton, VA 23365 |
| 1 | Director<br>National Aeronautics and Space Administration<br>Lewis Research Center<br>21000 Brookpark Road<br>Cleveland, OH 44135 |
| 1 | Director<br>Lawrence Livermore Laboratory<br>ATTN: Dr. M. Wilkins<br>P.O. Box 808<br>Livermore, CA 94550 |

| No. of Copies | Organization |
|---|---|
| 3 | University of Illinois<br>Aeronautical and Astronautical Engineering Department<br>101 Transportation Building<br>ATTN: Prof. A. R. Zak<br>Urbana, IL 61801 |

Aberdeen Proving Ground

Marine Corps Ln Ofc
Dir, USAMSAA